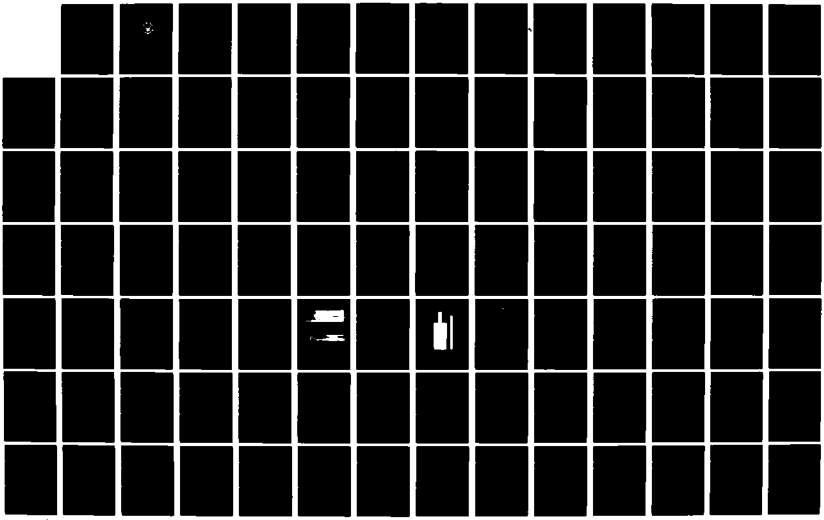
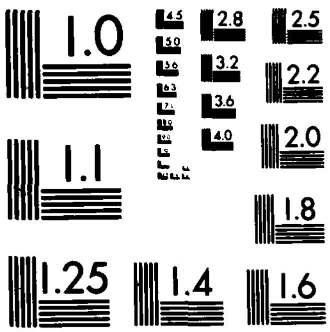
AD-A120 456 PRELIMINARY MEASUREMENTS AND CODE CALCULATIONS OF FLOW THROUGH A CASCADE OF DCA BLADING AT A SOLIDITY OF 167 (U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W D MOLLOY JUN 82

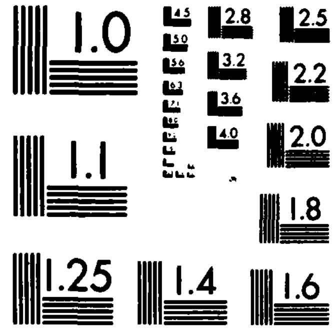
1/3

UNCLASSIFIED F/G 21/5 NL

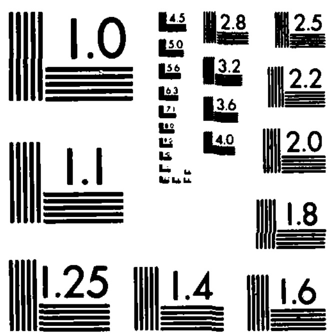




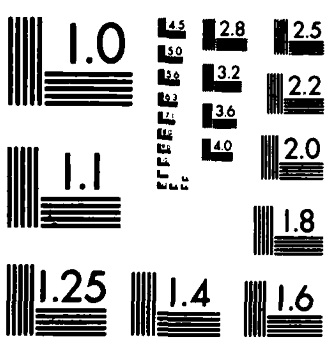
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



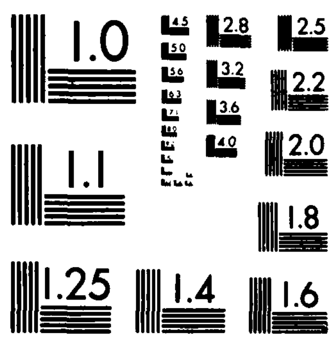
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PRELIMINARY MEASUREMENTS AND CODE CALCULATIONS OF FLOW THROUGH A CASCADE OF DCA BLADING AT A SOLIDITY OF 1.67

by

William D. Molloy Jr.

June 1982

Thesis Advisor: Raymond P. Shreeve

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. A120 456

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

Preliminary Measurements and Code Calculations of Flow through a Cascade of DCA Blading at a Solidity of 1.67

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis
June 1982

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

William D. Molloy Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

June 1982

13. NUMBER OF PAGES

280

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Cascade
DCA Blades
QSONIC

Double Circular Arc (DCA)

+ or -

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An experimental program to obtain uniform inlet flow to the test blading in a large cascade facility designed to use inlet turning vanes, and to measure the conventional blade element performance, is described. Attempts to reduce non-uniformities (<1% in velocity) using screens were unsuccessful and so abandoned. Preliminary DCA blade element performance data were obtained without screens at one

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

incidence angle before aero-mechanical problems with the inlet guide vane assembly curtailed testing. The blade surface pressure distribution at the one test condition compared very favorably with the distribution predicted using the NASA computer code QSONIC. Recommendations were made that would avoid the aero-mechanical problems encountered.

A

Approved for public release; distribution unlimited

Preliminary Measurements and Code Calculations of Flow through a Cascade of DCA Blading at a Solidity of 1.67

by

William D. Molloy Jr.
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1974

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1982

Author: W D Molloy Jr.

Approved by: Raymond P Shreeve
Thesis Advisor

Daniel J Collins
Chairman, Department of Aeronautics

William M Tolles
Dean of Science and Engineering

ABSTRACT

An experimental program to obtain uniform inlet flow to the test blading in a large cascade facility designed to use inlet turning vanes, and to measure the conventional blade element performance, is described. Attempts to reduce non-uniformities (±1% in velocity) using screens were unsuccessful and so abandoned. Preliminary DCA blade element performance data were obtained without screens at one incidence angle before aero-mechanical problems with the inlet guide vane assembly curtailed testing. The blade surface pressure distribution at the one test condition compared very favorably with the distribution predicted using the NASA computer code QSONIC. Recommendations were made that would avoid the aero-mechanical problems encountered.

TABLE OF CONTENTS

I.	INTRODUCTION	15
II.	FACILITY DESCRIPTION AND MEASUREMENT APPROACH	20
	A. SUBSONIC CASCADE WIND TUNNEL	20
	B. INSTRUMENTATION	22
	C. REFERENCE MEASUREMENTS	23
	D. TEST BLADING	23
	E. DATA ACQUISITION, REDUCTION AND ANALYSIS	24
III.	EXPERIMENTAL PROGRAM AND RESULTS	25
	A. PROGRAM OF TESTS	25
	B. TEST PROCEDURES	25
	1. Cascade Adjustments	25
	2. Measurements	27
	C. VERIFICATION OF INLET GUIDE VANE (IGV) ASSEMBLY	27
	D. TESTING WITH WIRE GAUZE SCREENS	28
	E. PRELIMINARY TESTING OF DCA BLADES	30
IV.	DISCUSSION OF EXPERIMENTAL RESULTS	31
	A. EFFECT OF INLET GUIDE VANE (IGV) MODIFICATION	31
	B. EFFECT OF WIRE GAUZE SCREENS	32
	C. PRELIMINARY TESTING OF DCA BLADES	34
	1. Inlet Uniformity	34
	2. Two-Dimensionality	34
	3. Periodicity	36

4. Blade Performance 36

5. Aero-Mechanical Problems Encountered . . . 37

V. COMPUTATIONAL PROGRAM 40

A. DESCRIPTION OF QSONIC 40

B. APPLICATION TO THE TEST CASCADE 42

C. COMPARISON OF CODE CALCULATIONS AND MEASURED DATA 43

VI. CONCLUSIONS AND RECOMMENDATIONS 44

APPENDIX A: MODIFICATION TO THE INLET GUIDE VANE SECTION OF THE SUBSONIC CASCADE WIND TUNNEL 119

APPENDIX B: SELECTION AND INSTALLATION OF SCREEN MATERIAL 126

APPENDIX C: CASCADE PERFORMANCE PARAMETERS (by F. S. Cina; reproduced with minor changes from Ref. 7) 130

APPENDIX D: INSTRUCTIONS FOR PREPARING INPUT AND OPERATING QSONIC USING A RECTILINEAR CASCADE CONFIGURATION 132

D.1. BACKGROUND INFORMATION 132

D.2. INPUT DESCRIPTION 134

D.3. PREPARING INPUT FILES 146

D.4. PROGRAM OUTPUT 150

D.5. RUNNING THE PROGRAM 151

D.6. QSONIC UPDATE 153

APPENDIX E: QSONIC PROGRAM LISTING 169

LIST OF REFERENCES 278

INITIAL DISTRIBUTION LIST 280

LIST OF TABLES

I.	MEASUREMENT UNCERTAINTY	47
II.	CASCADE CONFIGURATION FOR DCA BLADE TESTS	48
III.	SUMMARY OF MEASUREMENTS WITHOUT SCREENS	49
IV.	SUMMARY OF MEASUREMENTS WITH SCREENS ($\beta_1 = 35^\circ$) .	50
V.	PROBE DATA, UPPER PLANE AT MIDSPAN ($i = 5.3^\circ$) . .	51
VI.	PROBE DATA, LOWER PLANE AT MIDSPAN ($i = 5.3^\circ$) . .	52
VII.	CENTER BLADE DATA ($i = 5.3^\circ$)	53
VIII.	ADJACENT BLADES DATA ($i = 5.3^\circ$)	54
IX.	BLADE PERFORMANCE DATA	55
D.1.	INPUT DATA FOR MESH GENERATION RUN OF QSONIC . .	154
D.2.	TEST BLADE COORDINATES	155
D.3.	INPUT DATA FOR FLOW SOLUTION RUN OF QSONIC . . .	156
D.4.	DATA FILE FOR QUASI-3D SOLUTION	157
D.5.	SAMPLE OUTPUT FROM MESH GENERATION RUN	158
D.6.	SAMPLE OUTPUT FROM FLOW SOLUTION RUN	161

LIST OF FIGURES

1. Subsonic Cascade Facility 56

2. Plenum Chamber as Modified by Bartocci 57

3. Plenum Chamber as Modified by Moebius 58

4. Lower Plane Survey Probe 59

5. Upper Plane Survey Probe 59

6. Blade Edge Detail 60

7. Photograph of Instrumented Blade 61

8. Instrumented Blade 62

9. Instrumented Blade Tap Locations 63

10. Data Acquisition System 64

11. Probe Survey Data at Midspan of Lower Plane, End Walls at 35°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$. 65

12. Probe Survey Data at Midspan of Lower Plane, End Walls at 35°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 66

13. Probe Survey Data at Midspan of Upper Plane, End Walls at 35°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$. 67

14. Probe Survey Data at Midspan of Upper Plane, End Walls at 35°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 68

15. Probe Survey Data at Midspan of Lower Plane, End Walls at 30°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$. 69

16. Probe Survey Data at Midspan of Lower Plane, End Walls at 30°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 70

17. Probe Survey Data at Midspan of Upper Plane, End Walls at 30°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$. 71

18. Probe Survey Data at Midspan of Upper Plane, End Walls at 30°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 72

19. Probe Survey Data at Midspan of Lower Plane, End Walls at 50°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$. 73

20. Probe Survey Data at Midspan of Lower Plane, End Walls at 50°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 74

21. Probe Survey Data at Midspan of Upper Plane, End Walls at 50°, $(P_{PLENUM} - P_t)/Q_{ref}$ 75

22. Probe Survey Data at Midspan at Lower Plane, End Walls at 50°, Two Runs, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$ 76

23. Probe Survey Data at Midspan at Lower Plane, End Walls at 50°, Two Runs, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 77

24. Probe Survey Data at Midspan at Lower Plane, End Walls at 30°, Two Runs, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$ 78

25. Repetitive Samples with Fixed Probe Position (10" Left of CTR Midspan, End Walls 30°, Lower Plane, $(P_{PLENUM} - P_{AMB})/Q_{ref})$ 79

26. Repetitive Samples with Fixed Probe Position, 10" Left of CTR, Midspan, End Walls at 30°, Lower Plane $(P_t - P_{AMB})/Q_{ref}$ 80

27. Repetitive Samples with Fixed Probe Position, 10" Left of CTR, Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_t)/Q_{ref}$ 81

28. Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_{AMB})/Q_{ref})$ 82

29. Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_t - P_{AMB})/Q_{ref})$ 83

30. Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_t)/Q_{ref})$ 84

31. Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_{AMB})/Q_{ref})$ 85

32. Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_t - P_{AMB})/Q_{ref})$ 86

33. Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_t)/Q_{ref}$) 87

34. Probe Survey Data at Midspan, Lower Plane, 16 Mesh Screen, Walls at 35°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$ 88

35. Probe Survey Data at Midspan, Lower Plane, 16 Mesh Screen, Walls at 35°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 89

36. Probe Survey Data at Midspan, Upper Plane, 16 Mesh Screen, Walls at 35°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$ 90

37. Probe Survey Data at Midspan, Upper Plane, 16 Me Screen, Walls at 35°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 91

38. Probe Survey Data Span Traverse, Lower Plane, 16 Mesh Screen, Walls at 35°, 10" Left of CTR, $(P_{PLENUM} - P_t)/Q_{ref}$ 92

39. Probe Survey Data Span Traverse, Lower Plane, 16 Mesh Screen, Walls at 35°, Center of Test Section, $(P_{PLENUM} - P_t)/Q_{ref}$ 93

40. Plane Survey Data Span Traverse, Lower Plane, 16 Mesh Screen, Walls at 35°, Center of Test Section, $(P_{PLENUM} - P_t)/Q_{ref}$ 94

41. Probe Survey Data at Midspan, Lower Plane, 16 Mesh and 2 Mesh, Walls at 35°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$ 95

42. Probe Survey Data at Midspan, Lower Plane, 16 Mesh and 2 Mesh, Walls at 35°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 96

43. Probe Survey Data at Midspan, Lower Plane, 4 Mesh Screen, Walls at 35°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$ 97

44. Probe Survey Data at Midspan, Lower Plane, 4 Mesh Screen, Walls at 35°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 98

45. Probe Survey Data at Midspan, Lower Plane, 5 Mesh Screen, Walls at 35°, Points 1 to 50, $(P_{PLENUM} - P_t)/Q_{ref}$ 99

46. Probe Survey Data at Midspan, Lower Plane, 5 Mesh Screen, Walls at 35°, Points 51 to 100, $(P_{PLENUM} - P_t)/Q_{ref}$ 100

47. Wall Static Pressure Distribution 101

48. Probe Survey Data at Upstream Midspan (i = 5.3, $P_{PLENUM} - P_t)/\overline{Q}_1$, Lower Plane) 102

49. Blade Surface Pressure Distribution on Three Centermost Blades (i = 5.3) 103

50. Probe Survey Data at Midspan (i = 5.3, $(P_{PLENUM} - P_t)/\overline{Q}_1$, Upper Plane) 104

51. Probe Survey Data at Midspan (i = 5.3, $(X/\overline{X})$, Upper Plane) 105

52. Probe Survey Data at Midspan (i = 5.3, $(P_s - P_{w1})/\overline{Q}_1$, Upper Plane) 106

53. Probe Survey Data at Midspan (i = 5.3, Outlet Angle, Upper Plane) 107

54. Spanwise Probe Data Surveyed 1 in. from Suction Side of Centermost Blade (i = 5.3, $(P_{PLENUM} - P_t)/\overline{Q}_1$, Upper Plane) 108

55. Spanwise Probe Data Surveyed 1 in. from Suction Side of Centermost Blade (i = 5.3, $X/\overline{X}$, Upper Plane) . . 109

56. Spanwise Probe Data Surveyed 1 in. from Pressure Side of Centermost Blade (i = 5.3, $(P_{PLENUM} - P_t)/\overline{Q}_1$, Upper Plane) 110

57. Spanwise Probe Data Surveyed 1 in. from Pressure Side of Centermost Blade (i = 5.3, $X/\overline{X}$, Upper Plane) 111

58. Resultant Blade Force Vectors by Momentum Balance (– – – – –) and from Surface Pressure Integration (————), i = 5.3 112

59. Measured Blade Surface Pressure Distribution (i = 5.3, * = Pressure Side, + = Suction Side) 113

60. Measured Blade Surface Velocity Distribution (i = 5.3, * = Pressure Side, + = Suction Side) 114

61. 2D Code Blade Surface Mach Number Distribution (i = 5.3) 115

62. 3D Code Blade Surface Mach Number Distribution (i = 5.3) 116

63. Measured Blade Surface Mach Number Distribution (i - 5.3) 117

64. Blade Surface Mach Number Distribution (i = 5.3) . 118

A.1. Inlet Guide Vane Assembly 121

A.2. Cascade Wind Tunnel Sub-Assemblies 122

A.3. View of th IGV Adjustment Mechanism 123

A.4. Side View of the IGV Assembly 124

A.5. View of the Subsonic Cascade Wind Tunnel (North Wall Removed) 125

B.1. Screen Installation 129

D.1. Blade Coordinates 167

D.2. Blade Coordinates Translated and Rotated 167

D.3. Mesh Points on Blade Surface, Horizontal Chord . . 168

D.4. Mesh POints on Blade Surface, Chord at Stagger Angle 168

LIST OF SYMBOLS

AVDR	Axial velocity-density ratio
Cp_1	Coefficient of pressure at the inlet
Cp_2	Coefficient of pressure at the outlet
Cp_{STATIC}	Coefficient of static pressure rise
C	Blade chord (inches)
D	Diffusion factor
i	Incidence angle (degrees)
P	Pressure (in. H_2O)
Q	Dynamic pressure (in. H_2O)
T	Temperature ($^{\circ}R$)
X	Non dimensional velocity
β	Air angle, measured in the cascade midspan plane with respect to the axial direction (degrees)
γ	Stagger angle
σ	Solidity (C/S)
$\bar{\omega}$	Loss coefficient

Subscripts

amb	Ambient
P	Pressure
PLENUM	Plenum (supply)
s	Static
wl	North wall, lower plane

ACKNOWLEDGMENT

I would like to express my sincere thanks and appreciation to Dr. R. P. Shreeve, Director, Turbopropulsion Laboratory, for his guidance during the project. His affable nature and professional expertise made this study a most enjoyable learning experience.

A special note of thanks to Jim Hammer, Al McGuire, John Morris and Kelly Harris. Without their prompt and efficient help, this study would not have been possible.

I. INTRODUCTION

The need for lightweight, fuel efficient gas turbines that are capable of developing large amounts of thrust or power has motivated a continuing drive to obtain more accurate predictions of the flow through turbomachinery. Cascade testing of blade rows has, in the past, been a logical and relatively inexpensive way to learn more about the phenomena involved in the flow through compressor and turbine stages. It is required more today in order to verify two dimensional and near-two dimensional analysis codes for flow through cascades. Such testing also provides two-dimensional blade element performance data which, in the absence of reliable analytical predictions, are required in the design of compressors and turbine stages. Reference 1 describes how cascade measurements are obtained using a cascade wind tunnel and then used in the design process.

Before subsonic cascade wind tunnel data can be accepted as being valid, the flow conditions must meet three requirements. These criteria are discussed in detail in Refs. 1, 2 and 3. First, any disturbance in the airflow should be caused by the test blades; that is, the inlet flow to the test section must be acceptably uniform.

Secondly, the measured flow characteristics should, ideally, be independent of spanwise position along the test

blades. The flow, ideally, should be two dimensional. Duval [Ref. 3] demonstrated that excellent flow conditions could be achieved in the Naval Postgraduate School Turbopropulsion Laboratory (NPS/TPL) Subsonic Cascade Wind Tunnel using test blades with an aspect ratio of approximately two. The absence of suction along the walls results however, in some degree of streamline contraction which is measured in terms of an Axial Velocity Density Ratio (AVDR).

The third requirement which must be satisfied is the periodicity of the inlet flow to the test section and of the outlet flow. Within one chord length of the leading edges of the test blades an upstream perturbation occurs as the streamlines adjust to negotiate the blade passages. Since the rectilinear cascade is simulating an infinite cascade of blades, the flow characteristics should be the same at corresponding axial and blade-to-blade positions within each blade passage. This same condition should be true at any measurement plane downstream of the test blading.

As described by Rose and Guttormsen [Ref. 4] several unique features were incorporated into the design of the NPS/TPL Cascade to ensure a two-dimensional and periodic flow at the test blading. Initial evaluations of the facility were conducted and reported in Refs. 3, 4, 5 and 6. Work by Moebius [Ref. 6] involved modifications to the tunnel plenum chamber which established satisfactory uniform

flow at the exit of the bellmouth contraction into the test section.

In order to maintain an aspect ratio close to 2.0 at a solidity of 1.67 Cina [Ref. 7], following the work of Duval [Ref. 3], used a cascade configuration of 20 blades with 3 inch spacing. Cina conducted a program of tests of DCA blading at five (5) different air incidence angles. With this cascade configuration, Cina found that the inlet flow to the test section was uniform in direction and of uniform static pressure, but with an imposed variation in velocity and stagnation pressure resulting from the wakes of inlet guide vanes. Although excellent periodicity was found over pairs of test blades, departure from strictly periodic conditions were detected from one blade passage to another. Cina explained this condition as being the result of the inlet guide vane wakes being separated at two inch intervals and entering a test section configured with a three inch blade spacing. Because of these flow conditions, Cina considered his results to be preliminary.

As a result of these findings the Cascade Wind Tunnel was modified so that inlet guide vanes were provided at one inch intervals. The object of the study reported herein was to obtain blade performance data on Cina's cascade with good periodic and uniform flow conditions. A necessary condition was to obtain agreement in the results for blade forces evaluated from surface pressures and from a momentum

balance. A second objective was to compare measured blade surface Mach numbers with the results of code calculations.

At the outset, it was first necessary to carry out an extensive testing program to verify the new inlet guide vane section and the effect the new spacing had on flow uniformity and periodicity. It was found that the uniformity of dynamic pressure was improved with the inlet guide vanes spaced at one inch intervals. Attempts were made to further improve the flow by the use of (various) wire screens placed downstream of the inlet guide vanes. These methods proved unsuccessful for the range of parameters tested and in fact aggravated the situation.

Cina's testing of the Double Circular Arc blading was repeated without screens and with the Cascade Wind Tunnel configured with the modified inlet guide vane arrangement. Limited measurements were obtained before aero-mechanical problems with the new IGV arrangement, at the higher tunnel speeds, were encountered.

The overall purpose of the testing program initiated by Cina was to obtain data with which to verify design optimization computer codes developed by NASA. Towards this goal a fast, reliable computer analysis code (QSONIC) for calculating the flow field about a cascade of arbitrary 2-D airfoils was obtained from NASA. The code was adapted and modified to run on the Naval Postgraduate School's IBM 370/3033 computer.

The program QSONIC was developed by NASA to overcome the Mach number limitations of the earlier program TSONIC [Ref. 8]. QSONIC is described in Ref. 9. Procedures for using the program QSONIC at the Naval Postgraduate School's computer facility are given in Appendix D. The procedures are documented for the case of the DCA blading in the NPS/TPL cascade wind tunnel. A program listing is included to document changes made to the code in order to adapt to the operating system of the NPS computer.

Preliminary results show that experimental measurements and code predictions are in very good agreement.

II. FACILITY DESCRIPTION AND MEASUREMENT APPROACH

A. SUBSONIC CASCADE WIND TUNNEL

The Naval Postgraduate School's Rectilinear Cascade Facility is shown in Fig. 1. A description of the facility as it was originally configured is given in Ref. 4. The test facility is an open cycle wind tunnel, designed for the purpose of testing cascades of axial-flow turbomachinery compressor or turbine blades. The unique design of the test section ensures that the airflow paths from the inlet guide vanes to all of the blades of the cascade test section are of equal length. This particular design was intended to eliminate the problems found in other cascade wind tunnels caused by having wall boundary layers of different thicknesses entering the cascade at different points.

As a result of the work reported in Ref. 1, two fine mesh screens were installed at the bellmouth entrance to improve flow stability. A follow-on study into the cascade performance was conducted by Bartocci and is reported in Ref. 5. As a result of Bartocci's findings, plenum turning vanes were installed to direct plenum inlet air towards the bellmouth entrance and to decrease the total pressure fluctuations present at the bellmouth entrance. Figure 2 shows the configuration of the plenum chamber as modified by Bartocci. Reference 6 describes work by Moebius that resulted

in further modification to the plenum chamber in which the original contraction was changed to two two-dimensional contractions in series. After this modification, acceptably small variations in velocity and flow angle were measured at the inlet guide vane station. Figure 3 shows the internal arrangement of the plenum chamber as modified by Moebius and as it was configured for the work presented here.

Using the plenum configuration shown in Fig. 3, Duval [Ref. 3] found that the wakes from the inlet guide vanes were not mixed out at the lower measuring plane of the test section but gave a well defined periodic variation in the impact pressure. The peak-to-peak variation was ±4% of dynamic pressure over two-inch periodic intervals. This condition was undesirable, but was tolerated while looking only to establish the values of parameters required to achieve two-dimensionality and periodicity. Since the inlet flow conditions were not uniform, mass averages were used to calculate properties at the inlet plane from probe measurements.

In order to achieve a solidity of 1.67 and aspect ratio of about 2, a blade spacing of 3 inches was required for the tests carried out by Cina [Ref. 7]. The tests showed unacceptable departures from blade-to-blade periodicity under conditions of high blade loading and the installation of additional guide vanes was recommended. The modification

to the inlet guide vane section of the tunnel resulting from Cina's findings is described in detail in Appendix A.

In the present work, several tests were completed with the tunnel further modified by the introduction of wire screens between the inlet guide vanes and the lower plane of the test section. Appendix B describes the screen material and the criteria used to select the particular screens used in this study.

B. INSTRUMENTATION

The instrumentation used in the present study is that which is described in detail in Ref. 7. Twenty static pressure taps were located on the north and south side walls. The taps on the south wall were connected to a water manometer board so that the uniformity of the static pressure distribution of the inlet and outlet could be monitored visually. Additionally, one upstream tap on each wall and one downstream tap on each wall (near the centerline) were also connected to the Scanivalve so that these static pressures were recorded.

Figure 4 shows the probe that was used for the upstream survey (at the lower plane). The probe was a United Sensor Corporation DA 125 probe, serial number A847-1, calibrated earlier at various Mach numbers and yaw angles in a calibration facility. The United Sensor Corporation DC-125-24-F-22-CD probe, serial number A981-2 (Fig. 5), which was used

at the upper plane was similarly calibrated. The characteristics of the probes were approximated analytically to facilitate automatic data reduction procedures. The calibration and application procedures were those given by Duval [Ref. 3]. Appendix B of Ref. 3 describes both the upstream and downstream probes in detail. The mounting and traversing mechanisms are described in Ref. 7.

C. REFERENCE MEASUREMENTS

Plenum chamber (supply) pressure and temperatures, and atmospheric pressure were recorded on each data scan. Plenum pressure was also displayed on a water manometer board. The total temperature in the test cascade was assumed to be the same as the plenum chamber temperature.

D. TEST BLADING

The double circular arc test blading modeled the midspan section of the stator of the compressor stage reported in Ref. 10. Coordinates describing the profile of the blading are listed in Table D-2. The leading edge and trailing edge are shown in detail in Fig. 6. A photograph of the centermost blade is shown in Fig. 7.

The three blades centrally located in the cascade were constructed with surface pressure taps along the midspan section as shown in Fig. 8. The centermost blade had 19 ports on each of the pressure and suction surfaces and one tap at the leading edge. The two blades adjacent to the

center blade had 3 surface pressure taps located on each of the pressure and suction surfaces. The surface pressure tap locations for the centermost blade are given in Fig. 9.

E. DATA ACQUISITION, REDUCTION AND ANALYSIS

Data were recorded, reduced, and plotted using the modified Hewlett Packard HP-3052A Data Acquisition System shown in Fig. 10. Reference 11 describes the system in detail. The system incorporated a HP-9845A desktop computer as a controller, with all components connected on the HP-98034A HP-IB Interface Bus. A NPS/TPL HG-78K Scanivalve Controller with two 48 port Scanivalves allowed the programmed acquisition of probe and blade surface pressure measurements.

The software used in the present study for acquisition, reduction and plotting of data were developed from software originally created by Duval and Cina. The programs are listed and described separately in Ref. 12.

The uncertainties in the measurements are listed in Table 1.

III. EXPERIMENTAL PROGRAM AND RESULTS

A. PROGRAM OF TESTS

The test program was in three phases. First, in order to verify the new inlet guide vane assembly, tests were conducted with no blading in the test section and with the upper and lower endwalls set parallel at 35° (design condition), 30° and 50° with respect to axial.

Secondly, tests were made of the effect of wire gauze screen materials in reducing non-uniformities in the flow entering the test section. Appendix B describes the type of screens used and how they were installed.

The last phase of the test program was a continuation of the work initiated by Cina. Table II lists the cascade configuration tested. One test was completed successfully before aero-mechanical problems were encountered and testing was halted until the causes were analyzed.

B. TEST PROCEDURES

1. Cascade Adjustments

In the first and second phases of testing, the same procedures were used to realign the cascade for each new configuration. The lower and upper end walls were set to the desired flow angle and the inlet guide vanes were set so that their trailing edges were approximately aligned with

the end walls. The flow was started, and the desired inlet dynamic pressure was set. All tests were run at an average dimensionless inlet velocity (X) of about .13, corresponding to an inlet flow dynamic pressure of 18 inches water. Before recording data, the water manometer board was checked to ensure that the distributions of wall static pressures at the inlet plane and outlet plane were acceptably uniform. If required, the inlet guide vanes were adjusted to obtain uniform static pressure to within ±0.5 inches of water.

In the third phase of testing, initially the procedures used by Cina [Ref. 7] were followed, namely: the lower end walls were set to the desired inlet air angle and the upper end walls were set approximately to the expected exit air angle. The inlet guide vanes were set very approximately and the cascade was turned on and set to an inlet dynamic pressure of 18 inches water. The upper end walls and the inlet guide vanes were adjusted in turn to obtain wall static pressure distributions upstream and downstream which were acceptably uniform. Using this procedure however it was found on occasion that the inlet air angle sensed by the probe at the lower plane at mid-span could be 2 or 3 degrees different from the setting of the end walls.

The following procedures was subsequently adopted. The lower end walls were set to the desired inlet air angle. The upper end walls were adjusted to be "wide open", to form a diverging passage in which, when the cascade was turned

on (to an inlet dynamic pressure of 18 inches water), the flow was completely separated. The inlet guide vanes were adjusted to obtain the required inlet air angle on the channel center line over the center 24 inches in the blade-to-blade direction. The upper end walls were then moved individually towards the vertical until the lower plane static pressure distribution was uniform and the upper plane static pressure distribution was acceptably uniform at a value close to atmospheric pressure. No readjustment of the inlet guide vanes was made.

2. Measurements

Probe surveys were carried out in the blade-to-blade direction at midspan at the upper and lower planes. In the first and second phases of testing, data were taken over approximately 24 inches of the test section at intervals of 0.25 inches. Also, in order to test the repeatability of measurements, repetitive samples were taken with the probe held fixed at midspan at the lower plane at the center, 10 inches to the right and 10 inches to the left of center.

During the third phase of testing, data were taken using the procedures established by Cina in Ref. 7.

C. VERIFICATION OF INLET GUIDE VANE (IGV) ASSEMBLY

The results of the first phase are presented (as shown in Table III) in Figs. 11 to 32. The results are arranged into groups. The first group (Figs. 11 to 14) are

measurements of tunnel conditions with the end walls set at 35 degrees. Plots of conditions at the lower plane are followed by plots of conditions at the upper plane.

Results for a wall angle of 30 degrees are given next (Fig. 15 to Fig. 18), followed by results for a wall angle of 50 degrees (Fig. 19 to Fig. 21). Data were taken over 24 inches at the lower plane, and also at the upper plane at 30°. At 50°, at the upper plane, only the center 12 inches were surveyed.

The degree of repeatability of conditions in the wind tunnel from test to test (with no change in wall setting) is demonstrated by the results plotted in Figs. 22, 23 and 24.

The last group of plots, Figs. 25 to 33, shows the degree of repeatability in the probe data from scan to scan. Data for these plots were obtained by holding the probe stationary at midspan in three specific blade-to-blade locations in turn and taking 50 repetitive scans of the channels normally recorded for survey profile data. The time interval for each scan was approximately 20 seconds.

D. TESTING WITH WIRE GAUZE SCREENS

The selection and installation of the wire gauze screens is described in Appendix B.

The measurements obtained with the various screen configurations are given (as shown in Table IV) in Figs. 34 to 45. The results are arranged in four groups.

The first group of plots (Fig. 34 to 40) give data obtained with the 16 mesh .0105 inch diameter wire screen installed. Over a blade-to-blade distance from -1.0 to 7.0 inches (Fig. 35) a peak-to-peak variation in velocity of about 1 percent was noted. This is slightly greater than the less than 1 percent (0.9 percent) variation noted over the same survey region without a screen installed (Fig. 12).

The results shown plotted in Figs. 41 and 42 were obtained with two screens installed. One screen (16 mesh, .0105 inch diameter) was installed as discussed in Appendix B, while the second screen (2 mesh, .0400 inch diameter) was attached across the duct at the leading edges of the inlet guide vanes.

The fourth group of plots (Figs. 43 to 46) show the results for two different single screens. Probe survey data for these screens was taken only at the lower plane. The results shown plotted in Figs. 43 and 44 are data obtained with a 4 mesh, .041 inch diameter wire screen installed. The variation in velocity in the blade-to-blade direction was as much as ±1.1 percent, peak-to-peak. The results shown in Figs. 45 and 46 are for a 5 mesh, .041 inch diameter wire screen. The variation in flow velocity was approximately ±1.5 percent, peak-to-peak.

E. PRELIMINARY TESTING OF DCA BLADES

The results contained in Tables V to IX and Figs. 47 to 60 are arranged in the following manner.

The results shown plotted in Figs. 47 to 60 are divided into two separate groups. The first group (Figs. 47 to 57) contain results which exhibit the quality of the wind tunnel flow conditions. The second group (Figs. 58 to 60) shows the blade forces (and surface pressures) from survey data. In the first group of figures, results are presented first to examine the inlet flow uniformity (Figs. 47 and 48); second, to examine the outlet flow periodicity (Figs. 49 to 53); and, finally, to examine outlet flow two dimensionality (Figs. 54 to 57).

All points are shown connected with straight lines.

IV. DISCUSSION OF EXPERIMENTAL RESULTS

A. EFFECT OF INLET GUIDE VANE (IGV) MODIFICATION

The probe survey data shown in Figs. 11 and 12 were taken at the lower plane, with the end walls set at 35 degrees. A turning angle of 35 degrees corresponded to the "design point" of the inlet guide vanes, when the airflow from the plenum chamber was at zero angle of incidence to the leading edge of the IGV's. These two figures show that the inlet plane total pressure at midspan in the blade-to-blade direction had a peak-to-peak periodic variation of about ±2 percent and therefore about a ±1 percent peak-to-peak variation in the velocity. Corresponding data from probe surveys at the upper plane (Figs. 13 and 14) show that periodic variations in total pressure were reduced to about 25% of the value at the lower plane by the mixing of the inlet guide vane wakes.

It can be seen in Figs. 15 to 21 that at "off-design" conditions for the IGV's (endwalls at 30° and 50°) there is a greater periodic variation in total pressure at the lower plane in the blade-to-blade direction than at the design point conditions. In Fig. 15 and Fig. 16, it can be seen that the periodic variations in total pressure are more pronounced with the flow from the plenum at a negative incidence angle to the IGV's (endwalls at 30°). Except for the first

8 data points in Fig. 15, there is a well defined period of about 1 inch of travel.

Figures 19 and 20 show the probe surveys conducted with endwalls at 50 degrees. At this off design condition the periodic variation in total pressure was considerably greater than the design point and the individual wakes from the inlet guide vanes were much less well defined.

The repeatability of the survey was examined at wall angles of both 50^{o} and 30^{o}. Figs. 22 to 24 show that the non-uniformities in the flow conditions were repeated to (generally) better than 0.5 percent of total pressure. The question was then, to what accuracy could the individual data points be repeated in successive samples. This was examined at several probe positions and the results given in Figs. 25 to 33 explain the departures in Figs. 22 to 24.

B. EFFECT OF WIRE GAUZE SCREENS

All testing with screens was conducted with the end walls and inlet guide vanes set to yield a flow angle of 35 degrees. The data obtained with screens installed were therefore compared with the data obtained without screens, shown in Figs. 11-14. The effect of the pressure drop across the screen on the pressure coefficient plotted in Figs. 34-35 should be noted. With the first screen installed the drop in total pressure from plenum to the probe in the lower plane was about 10 inches of water (plenum pressure minus total

pressure measured by the probe at the lower plane). Without the screen (at design conditions), the pressure drop from the plenum to the probe at the lower plane was approximately 2.0 inches of water. Since Q_{ref} was defined as the difference between plenum pressure and lower wall static pressure, the value of Q_{ref} with the first screen installed was about 28 inches water and without the screen installed, about 20 inches water. In comparing the peak-to-peak variation in P1 seen in Fig. 11 with that obtained with the first screen installed in Fig. 34, the difference in the values of Q_{ref} must be considered. Examinations showed that the peak-to-peak variation in velocity remained at approximately 1% when the screen was installed.

Figures 38 to 40 are plots of data obtained during a spanwise traverse of the probe at the lower plane. These figures show that the pressure drop through the screen and turning vanes was nearly uniform over approximately 8.0 inches of the 10.0 inch span of the tunnel.

In an attempt to generate upstream disturbances that might trigger early boundary layer transition on the IGV's and increase the rate of mixing of the wakes, a second screen was attached to the leading edge of the IGV's. The results in Figs. 41 and 42 showed that this was not the case and in fact the second screen increased the magnitude of the non-uniformities at the lower survey plane. Measurements made with a single 4 mesh (Figs. 43 and 44) and a single 5 mesh

screen (Figs. 45 and 46) of similar blockage showed that neither screen influenced the flow in a particularly favorable manner. The 4 mesh screen caused the peak-to-peak variation in velocity to be about ±1.1 percent, while the 5 mesh screen caused the variation to be about ±1.5 percent. This compared unfavorably with the variation obtained without any screen installed which was less than ±1 percent.

It was therefore decided to proceed with measurements of the test cascade without using screens.

C. PRELIMINARY TESTING OF DCA BLADES

1. Inlet Uniformity

The probe survey at the lower plane in Fig. 48 shows that the inlet plane total pressure at midspan varied in the blade to blade direction less than 0.5 inches of water, with no well-defined spatial period. This was an improvement in the inlet conditions found by Cina [Ref. 7: Fig. 16]. That the spatial period was not well defined agreed with the findings presented earlier in this report. The wall static pressure distribution (Fig. 47) showed small variations (less than 0.5 ins. water peak-to-peak at the lower plane, .4 ins. water peak-to-peak at the upper plane).

2. Two-Dimensionality

The data in Figs. 54 to 57 show that, at the downstream plane, an area of (spanwise) nearly uniform conditions

existed near the centerline of the cascade. Reference 2 points out that at higher loadings it is difficult to establish a substantial spanwise area of uniform flow in the region near the suction side of the blade. This difficulty is evident in the data shown in Figs. 54 and 55 which show that only about 20% of the spanwise distance is acceptably uniform. It is noted that Cina also found reduced areas of uniform flow at this incidence angle; however, 30-40% of the spanwise distance was found to be acceptably uniform in his case. The difference could be the result of the reduced spacing of the IGV's and its effect on the side wall boundary layers.

Figure 58 shows results for inlet and outlet flow angles and blade force vectors derived in two ways as shown in Appendix B of Ref. 7. These two methods are first the applications of momentum conservation to probe survey data and second, the integration of surface pressures measured over the blade area. Reference 2 points out that for truly two dimensional flow the blade forces derived from the two methods should be the same. As shown in Fig. 58 the magnitudes and the directions of the two vectors representing the blade forces are in reasonable agreement. It is noted however that at this particular incidence angle Cina [Ref. 7] measured blade forces that were in total agreement in direction but disagreed slightly in magnitude. The values of the

force magnitudes were about 1.5% lower than those measured by Cina.

3. Periodicity

As can be seen in Figs. 50 and 51 the total pressure and velocity qualitatively repeated fairly well over three central blade passages. Acceptably small quantitative differences are noted. There was also a small but measurable difference in the surface pressures on adjacent blades, as is evidenced in Fig. 49.

4. Blade Performance

Figures 59 and 60 are plots of the pressure and velocity distributions respectively over the centermost blade. These results compare favorably to those obtained by Cina for an incidence angle of 5.3°.

Table IX contains the blade performance parameters deduced from the probe survey data listed together with the data obtained for corresponding test parameters in Ref. 7. While differences in two sets of data are evident, the differences are not large. It is noted that the value of the loss coefficient was only 10% lower than was measured by Cina, but the AVDR was less than 2% different from unity rather than the 6.5% measured by Cina. Further measurements need to be made, particularly in the light of the following discussion, before stronger conclusions can be drawn.

5. Aero-Mechanical Problems Encountered

Fifteen cascade tests were made while evaluating the new inlet guide vane assembly and testing the wire screens. All runs were made without test blades installed, with a plenum total pressure of about 20 inches of water and with the upper and lower end walls parallel. No difficulties were encountered in establishing the desired flow conditions or in using the inlet guide vanes to arrive at a satisfactory distribution of wall static pressures.

The first time the Cascade Wind Tunnel was set up with test blades installed to take data at an air inlet angle of 39.2°, the tunnel operated normally and the test was completed. (The data from this test were subsequently found to be highly suspect and are not reported here.) During the next test, with an inlet air angle of 42.4, the start-up appeared normal and previously established procedures were used to arrive at a satisfactorily uniform wall static pressure distribution. Tunnel operation appeared to be normal while taking data, but on shutdown a very noticeable high frequency vibration was encountered. Examination of the inlet guide vanes revealed that about 40% of the 60 blades were damaged. Damage included chips missing from the trailing edges, blades bent, cracks at the weld where the blade is joined to its support and indications that the suction side near the leading edge of one blade had been vibrating

against the pressure side near the leading edge of an adjacent blade.

After the inlet guide vanes had been repaired and reinstalled extreme care was used at the beginning of the next test to adjust the inlet guide vanes, with two individuals monitoring the movement of the adjustment mechanism. (IGV adjustment mechanism is described in detail in Appendix A.) A lack of stiffness in the mechanism was suspected as having been a contributing factor to the failure.

One successful test was completed at an inlet flow angle of 42.4° and these data were discussed above.

With the next cascade configuration set, at an air inlet angle of 45.9°, when the IGV's were adjusted after starting up, high frequency vibrations were again experienced. The wind tunnel was shut down and no further testing was attempted at plenum total pressures as high as 20 inches of water gauge.

The difficulty encountered with the IGV's is not fully understood, however the lack of stiffness present in the actuation of the two separate rows of vanes is suspected of having allowed the problem to occur. Certainly, the possibility of an aerodynamic flutter condition being present (due to the misadjustment perhaps) can not be ignored. It was noticed after the initial failure that the lead screw which adjusts the IGV's could be turned but the blades mounted from only one side would be caused to rotate. This

could lead to the trailing edge of one blade contacting the trailing edge of an adjacent blade and effectively closing the blade passages.

Also, the holes in which the cylindrical shanks of the IGV's were held were found not to be uniformly machined. As much as 0.1 inches of movement at the tip of some vanes was possible while others could barely move. (The most seriously damaged blades were found in or adjacent to the larger holes.)

The tendency for the mechanism to "hang-up" on one side would be greater as the vanes became more highly loaded. It is noted that the IGV problem was encountered first when going to increased incidence angles with the compressor test cascade installed. In setting a constant plenum total pressure of 20 inches of water gauge, the static pressure increase across the test blades to a constant atmospheric pressure at the downstream side implies that a progressively increasing dynamic pressure was being generated out of the turning vanes. This can be seen in Table IX, where Q_1 for the test at $\beta_1 = 42.4$ was 25 inches of water.

V. COMPUTATIONAL PROGRAM

A. DESCRIPTION OF QSONIC

The computational code, QSONIC, was developed by the staff at the NASA Lewis Research Center.[1] This code is able to calculate the blade-to-blade flow conditions in turbomachinery blade rows assuming inviscid flow but including streamtube convergence and radius change in the throughflow direction. QSONIC is flexible enough to allow the input of the appropriate boundary conditions to calculate the flow through the test blading in the Subsonic Cascade Wind Tunnel. The program uses a fully conservative solution of the full potential equation combined with the finite volume method on a body-fitted mesh. QSONIC uses an artificial density imposed in the transonic region, if such a region exists, to ensure stability and the capture of shock waves.

The analysis used by QSONIC is a combination of transonic analysis methods to calculate the flow conditions in the vicinity of a cascade of airfoils. A conservative form of the full potential equation is discretized at every point of a body fitted periodic mesh and a mass balance is calculated through the finite volume surrounding the point. The volume

[1]The help and advice received from Charles Farrell at NASA Lewis R.C. in the process of adapting QSONIC to the NPS computer is gratefully acknowledged.

is corrected three dimensionally for any change in streamtube thickness along a streamtube, if a quasi-3D solution is desired. Either elliptic or hyperbolic non-linear partial differential equations are used, depending on the local Mach number.

The analysis used in developing QSONIC made the following assumptions:

1) The airflow is inviscid and adiabatic.

2) The airflow relative to the test blades is steady.

3) Air is a perfect gas with constant specific heat.

4) The airflow is isentropic and any discontinuities such as shocks are so weak that they may be approximated as isentropic jumps.

5) There is no velocity component normal to the streamsurface.

6) The airflow relative to a fixed reference frame (i.e. absolute velocity) is completely irrotational.

Assumption 4 requires that the peak local relative Mach number on a blade surface be 1.4 or less. The Mach numbers measured in test blades in the Subsonic Cascade Wind Tunnel would be well within this limit. However, this limitation would probably preclude the use of QSONIC for analysis of the flow field in the NPS transonic cascade wind tunnel.

There are some combinations of blading geometry and flow conditions which cause unsatisfactory results to be generated. For example, because of assumptions 1 and 6, sharp leading edges at high incidence angles (more that a few degrees) cause large velocity peaks in the blade surface as

the flow tries to turn from the stagnation point to the suction surface.

Reference 9 gives a detailed description of QSONIC and the solution method used including the governing equations. Appendix D describes the operating procedures to use QSONIC on the Naval Postgraduate School's IBM 370/3033 computer. Appendix D also describes the input and output required as applicable to the Subsonic Cascade Wind Tunnel.

B. APPLICATION TO THE TEST CASCADE

Appendix D describes in detail the generation of the input required for QSONIC when applied to test blading in the Subsonic Cascade Wind Tunnel facility. In the present work one comparison of code calculations and measured data was made before testing was stopped. The comparison was for an inlet flow angle (β_1) of 42.4°.

Tables D.1 and D.3 show the input data generated. Table D.6 shows the flow solution output by QSONIC. The flow calculated on the blade surface, using a 15 by 97 mesh, was examined. Figure 61 is a plot of the calculated Mach number along the blade surface using two dimensional inputs. Figure 62 is a plot of computed Mach number incorporating quasi-three dimensional effects. The method of incorporating quasi-three dimensional effects is explained in Appendix D.

C. COMPARISON OF CODE CALCULATIONS AND MEASURED DATA

Table VII lists the data measured in the cascade wind tunnel. C_{p1}, C_{p2} and X_{vel} are defined in Appendix C. The surface Mach number distribution measured on the center blade is shown plotted in Fig. 63.

For comparison, the computed two dimensional, computed quasi-three dimensional, and the Mach number measured in the cascade wind tunnel are plotted together in Fig. 64.

Excellent agreement between all three cases is seen. As would be expected, the greatest difference between measured and calculated data is near the leading edge in the suction side and at the trailing edge of the blade.

VI. CONCLUSIONS AND RECOMMENDATIONS

Based on the results of the first part of the present study, to evaluate the effects of the altered inlet guide vane spacing on flow uniformity and periodicity, the following conclusions were drawn:

1. With the inlet guide vanes operating at design, the peak-to-peak variation in velocity was about ±1%, and there was a well defined spatial period of about 1 inch.

2. Operating the cascade wind tunnel in a configuration that requires the inlet guide vanes to be set to other than zero incidence resulted in peak-to-peak variations in velocity greater than ±1% and a spatial period that was less well defined.

The second part of the study, to evaluate the use of wire gauze screens to further reduce the non-uniformities in the flow field, led to the following conclusions:

1. A 16 mesh screen with a blockage factor of .69 had a slightly aggravating effect in the variation in velocity. The peak-to-peak variation in velocity with the screen installed was slightly greater than with no screen installed. This occurred at the expense of a pressure drop of about 10" of water across the screen.

2. The use of screens with similar blockage but with larger mesh and larger diameter wires resulted in larger peak-to-peak variations in the velocity at the lower plane.

The overall objective of the present study was to measure the performance of the DCA test blading. Because of aero-mechanical problems encountered with the inlet turning vanes the performance of the blades was obtained at one

incidence angle only. The following was concluded from the limited test program:

1. As a result of the reduced inlet guide vane spacing the variations in velocity and total pressure at the inlet plane were much less than those reported by Cina.

2. Good periodicity was found from one blade passage to another.

3. An acceptable region of spanwise uniformity (20-40% of blade span) was found at the downstream plane at the one test condition reported. However, this was less than was previously reported for the same incidence angle.

4. The blade forces derived from the integration of surface pressure measurements and probe survey data were in close agreement in both magnitude and direction.

5. The Mach number measured by surface pressure taps over the surface of the blade and Mach number calculated using the program QSONIC were in excellent agreement qualitatively and reasonable agreement quantitatively.

6. The specific reasons for the aero-mechanical problems experienced with the inlet turning vanes have not been identified completely.

Based on these conclusions and other observations, the following recommendations are made:

1. Use of the present inlet guide vane assembly and adjustment mechanism for testing at inlet dynamic pressures higher than about 15 inches of water is unsafe. There are three possible solutions to this problem.

   a) Operate only within the dynamic pressure range of 10-15 inches of water.

   b) Modify the new inlet guide vane assembly so that the vanes are supported at both ends on their axes of rotation. [Supporting the IGV's from both ends would prevent flapping vibrations of the (presently) cantilevered vanes. Such vibrations, when the vanes are supported alternately from opposite ends and the gaps are small compared to the chord, might lead to a potentially destructive flutter mode at particular flow velocities.]

c) Replace the entire inlet turning vane section with one of entirely new design.

2. The procedure should be adopted immediately of adjusting the vanes and walls of the cascade at lower values of the dynamic pressure before increasing the blower speed to the desired operating condition.

3. More time needs to be spent to examine the flow field produced between the guide vanes and the test blades, and to establish the effects of the movement of the tail boards. The uniqueness of the flow field when the wall static pressure uniformity is used as a criterion of good inlet flow, needs to be examined by conducting repetitive tests at nominally similar test parameters. Only when the adjustment of the flow in the facility and the quality of the flow itself is fully understood should the measured blade performance data be accepted as final.

4. The upper electrical yaw adjustment mechanism should be replaced with a manual system to greatly decrease the time required to achieve probe pressure (angle) balancing.

5. Develop the computer code necessary to take advantage of the plotting data created by QSONIC.

6. Modify the data acquisition and reduction software for the HP 9845 so that real time plots of blade performance parameters can be displayed.

TABLE I. MEASUREMENT UNCERTAINTY

Item	Description	Method	Uncertainty
x	Blade-to-Blade dimension x = 0 in. West end x = 60 in. East end	Position Potentiometer	±.01 in.
z	Spanwise dimension z = 0 in. North wall z = 10 in. South wall	Position Potentiometer on probe mount	±.01 in.
β_1	Inlet flow yaw angle	Angle Potentiometer on probe mount (hand adjustment)	±.2 deg.
β_2	Outlet flow yaw angle	Angle Potentiometer on probe mount (motor driven adjustment)	±.5 deg.
P_{plen}	Plenum total pressure	Static tap in plenum chamber $V \simeq 0$	±.01 in. H_2O gauge
P_s	Static pressure at the test plane	Calibrated pneumatic probe	±.1 in. H_2O gauge
$P_{w\ell}$	Static pressure at x = 0 in., y = -16.25 in., z = 0 in.	Static tap on North wall	±.01 in. H_2O gauge
P_{ATM}	Atmospheric pressure	Absolute Strain Gauge Transducer	±.3 in. H_2O
P	Pressure	Scanivalve Transducer	±.01 in. H_2O gauge

TABLE II. CASCADE CONFIGURATION FOR DCA BLADE TESTS

<u>Constant Parameters</u>

Number of Blades	20
Spacing (Pitch)	3 inches
Chord	5.01 inches
Solidity	1.67
Thickness	7.0 percent of chord
Camber Angle	45.72 degrees
Stagger Angle	14.72 degrees

<u>Variable Parameters</u>

β_1	42.4 degrees
i	5.3 degrees

TABLE III. SUMMARY OF MEASUREMENTS WITHOUT SCREENS

β_1	Survey Plane	Survey Direction	Fig. Nos.	Purpose
35	Lower	B-B (24 inches)	11 & 12	Flow Field Determination
	Upper	B-B (24 inches)	13 & 14	
30	Lower	B-B (24 inches)	15 & 16	
	Upper	B-B (24 inches)	17 & 18	
50	Lower	B-B (24 inches)	19 & 20	
	Upper	B-B (12 inches)	21	
50	Lower	B-B (24 inches)	22 & 23	Survey Repeatability
30	Lower	B-B (24 inches)	24	
30	Lower	Fixed Probe (10" L. of ℄)	25-27	Point Repeatability
		(on ℄)	28-30	
		(10" R. of ℄)	31-33	

TABLE IV. SUMMARY OF MEASUREMENTS WITH SCREENS
($\beta_1 = 35^\circ$)

	Screen	Survey Plane	Survey Direction	Fig. Nos.
1.	16 mesh .0105 wire	Lower	B-B	34 & 35
		Upper	B-B	36 & 37
		Upper	Spanwise	38-40
2.	16 mesh .0105 wire + 2 mesh ahead of IGV's	Lower	B-B	41 & 42
3.	4 mesh .041 wire	Lower	B-B	43 & 44
4.	5 mesh .041 wire	Lower	B-B	45 & 46

TABLE V. PROBE DATA, UPPER PLANE AT MIDSPAN (i = 5.3°)

DATA FROM FILE UBRED2:T14
BLADE TO BLADE TRAVERSE MIDSPAN

UPPER PLANE

Point	Loc(in)	Beta	Q/Q1bar	Ps/Q1bar	Pt/Q1bar	M/Mbar
1	-7.31	-1.59	.5686	.2614	.0920	.6475
2	-6.84	-.84	.5532	.2669	.1041	.6331
3	-6.37	-.83	.4276	.2561	.2396	.5630
4	-5.87	-2.56	.5096	.2707	.1380	.6149
5	-5.67	-2.56	.5622	.2638	.0995	.6428
6	-5.48	-2.56	.5780	.2534	.0902	.6530
7	-5.23	-1.79	.5739	.2573	.0916	.6503
8	-5.04	-1.79	.5747	.2626	.0927	.6483
9	-4.80	-1.78	.5697	.2573	.0952	.6480
10	-4.60	-1.78	.5705	.2588	.0901	.6495
11	-4.41	-1.78	.5687	.2575	.0955	.6478
12	-4.16	-1.36	.5650	.2591	.0994	.6451
13	-3.96	-1.35	.5629	.2606	.0987	.6443
14	-3.78	-1.36	.5462	.2682	.1123	.6333
15	-3.58	-1.36	.4863	.2684	.1739	.5978
16	-3.39	-1.35	.4349	.2611	.2269	.5677
17	-3.19	-1.88	.4239	.2571	.2456	.5597
18	-3.00	-3.16	.4886	.2738	.1668	.5989
19	-2.80	-2.41	.5467	.2744	.1108	.6318
20	-2.60	-2.19	.5717	.2636	.0962	.6461
21	-2.40	-1.92	.5751	.2608	.0951	.6482
22	-2.20	-1.92	.5773	.2597	.0926	.6498
23	-2.01	-1.92	.5814	.2599	.0926	.6507
24	-1.81	-1.92	.5810	.2612	.0951	.6493
25	-1.61	-1.48	.5765	.2606	.0901	.6502
26	-1.42	-1.47	.5755	.2644	.0912	.6483
27	-1.22	-1.48	.5748	.2626	.0947	.6476
28	-1.02	-1.46	.5679	.2661	.1004	.6430
29	-.83	-1.47	.5378	.2726	.1217	.6267
30	-.64	-1.46	.4704	.2693	.1954	.5862
31	-.44	-1.47	.4140	.2553	.2698	.5497
32	-.25	-1.47	.4295	.2658	.2442	.5594
33	-.05	-3.00	.5011	.2740	.1616	.6040
34	.15	-2.32	.5562	.2718	.1040	.6371
35	.36	-1.74	.5776	.2672	.0879	.6488
36	.55	-1.71	.5795	.2636	.0865	.6509
37	.75	-1.83	.5880	.2591	.0872	.6541
38	.95	-1.80	.5915	.2613	.0833	.6554
39	1.15	-1.83	.5902	.2591	.0844	.6555
40	1.35	-1.83	.5903	.2587	.0833	.6561
41	1.85	-1.83	.5810	.2601	.0922	.6506
42	2.35	-1.83	.4490	.2598	.2202	.5749
43	2.85	-1.84	.4421	.2670	.2278	.5681
44	3.34	-2.01	.5722	.2600	.0990	.6465
45	3.85	-2.01	.5763	.2597	.0972	.6481

TABLE VI. PROBE DATA, LOWER PLANE AT MIDSPAN (i = 5.3°)

DATA FROM FILE LBRED2:T14
BLADE TO BLADE TRAVERSE MIDSPAN

LOWER PLANE

Point	Loc(in)	Beta	Q/Q1bar	Ps/Q1bar	Pt/Q1bar	X/Xbar
1	-4.00	-42.43	.9533	-.1372	.0932	.8457
2	-3.50	-42.42	.9603	-.1444	.0967	.8474
3	-3.00	-42.42	.9567	-.1427	.0996	.8454
4	-2.50	-42.44	.9614	-.1519	.1031	.8480
5	-2.00	-42.43	.9741	-.1469	.0871	.8524
6	-1.50	-42.43	.9767	-.1493	.0875	.8532
7	-1.00	-42.43	.9722	-.1481	.0889	.8522
8	-.50	-42.43	.9743	-.1564	.0921	.8546
9	0.00	-42.43	.9799	-.1567	.0935	.8540
10	.50	-42.42	.9849	-.1592	.0854	.8585
11	1.00	-42.43	.9814	-.1604	.0850	.8593
12	1.50	-42.43	.9871	-.1628	.0779	.8635
13	2.00	-42.43	.9787	-.1615	.0890	.8582
14	2.50	-42.43	.9742	-.1632	.0971	.8556
15	3.00	-42.41	.9731	-.1650	.1059	.8526
16	3.50	-42.42	.9807	-.1663	.0924	.8589
17	4.00	-42.44	.9736	-.1651	.0968	.8567

TABLE VII. CENTER BLADE DATE (i = 5.3°)

X/C	Y/C	Cp1	Cp2	Mach	Nvel
PRESSURE SIDE CENTER BLADE					
.0007	.0054	.6655	.4573	.1785	.0796
.0160	.0019	.6417	.4140	.1849	.0824
.0319	.0066	.5240	.1999	.2136	.0951
.0479	.0112	.5287	.2083	.2126	.0946
.0858	.0215	.4871	.1327	.2219	.0988
.1218	.0303	.4818	.1230	.2231	.0993
.1956	.0452	.4700	.1017	.2257	.1004
.2695	.0576	.4757	.1120	.2244	.0999
.3433	.0663	.4764	.1133	.2243	.0998
.4192	.0716	.4892	.1366	.2215	.0986
.4930	.0736	.4871	.1327	.2219	.0988
.5669	.0727	.4771	.1146	.2241	.0997
.6407	.0678	.4956	.1482	.2201	.0979
.7146	.0601	.4889	.1359	.2216	.0986
.7884	.0487	.4949	.1469	.2202	.0980
.8283	.0411	.4672	.0965	.2263	.1007
.8683	.0327	.4537	.0719	.2292	.1020
.9082	.0230	.4245	.0189	.2354	.1047
.9481	.0123	.3815	-.0594	.2443	.1086
.9880	.0006	.2778	-.2482	.2646	.1175
SUCTION SIDE CENTER BLADE					
.0160	.0227	-1.3641	-3.2355	.4973	.2171
.0319	.0310	-.7756	-2.1647	.4249	.1867
.0479	.0389	-.4980	-1.6597	.3878	.1709
.0858	.0563	-.3541	-1.3979	.3675	.1622
.1218	.0710	-.3431	-1.3778	.3659	.1615
.1956	.0970	-.2923	-1.2853	.3585	.1583
.2695	.1170	-.2500	-1.2084	.3522	.1556
.3433	.1309	-.2038	-1.1243	.3453	.1526
.4192	.1399	-.1540	-1.0338	.3377	.1493
.4930	.1432	-.0854	-.9090	.3270	.1447
.5669	.1412	-.0584	-.8599	.3227	.1428
.6407	.1339	.0102	-.7351	.3116	.1380
.7146	.1209	.0631	-.6387	.3028	.1342
.7884	.1021	.1456	-.4887	.2886	.1280
.8283	.0895	.1911	-.4059	.2805	.1245
.8683	.0755	.2326	-.3303	.2730	.1212
.9082	.0593	.2636	-.2740	.2673	.1187
.9481	.0407	.2842	-.2365	.2634	.1170
.9880	.0206	.2948	-.2171	.2613	.1161

TABLE VIII. ADJACENT BLADES DATA (i = 5.3°)

X/C	Y/C	Cp1	Cp2	Mach	Vel
PRESSURE SIDE LEFT BLADE					
.1218	.0303	.4014	-.0231	.2402	.1068
.4192	.0716	.4619	.0868	.2275	.1012
.8283	.0411	.4491	.0635	.2302	.1024
SUCTION SIDE LEFT BLADE					
.1218	.0710	-.3438	-1.3791	.3660	.1615
.4192	.1400	-.1505	-1.0273	.3371	.1491
.8283	.0895	.1975	-.3943	.2794	.1240
PRESSURE SIDE RIGHT BLADE					
.1218	.0303	.4640	.0907	.2270	.1010
.4192	.0716	.4658	.0939	.2266	.1008
.8283	.0411	.4597	.0829	.2279	.1014
SUCTION SIDE RIGHT BLADE					
.1218	.0710	-.3285	-1.3513	.3638	.1606
.4192	.1400	-.1519	-1.0299	.3374	.1492
.8283	.0895	.1911	-.4059	.2805	.1245

TABLE IX. BLADE PERFORMANCE DATA

	Present Results	From Ref. 7
β_1	42.42	42.42
i	5.3	5.3
β_2	1.85	0.4
δ	10.44	9.0
D	0.455	0.46
$\overline{\omega}$	0.037	0.041
$\dfrac{\omega \cos^3 \beta_2}{2 \sigma \cos^2 \beta_1}$	0.020	0.023
$\dfrac{\omega \cos \beta_2}{2\sigma}$ $(\mathrm{x}\ 10^2)$	1.09	1.242
AVDR	1.015	1.065
$C_{P_{STATIC}}$	0.413	0.351
C_{xM}	-1.385	-1.380
C_{yM}	-0.669	-0.566
C_{xB}	-1.330	-1.476
C_{yB}	-0.572	-0.645
$\overline{Q}_1$ (in. H_2O)	25	22
$\overline{X}$	.14	.12

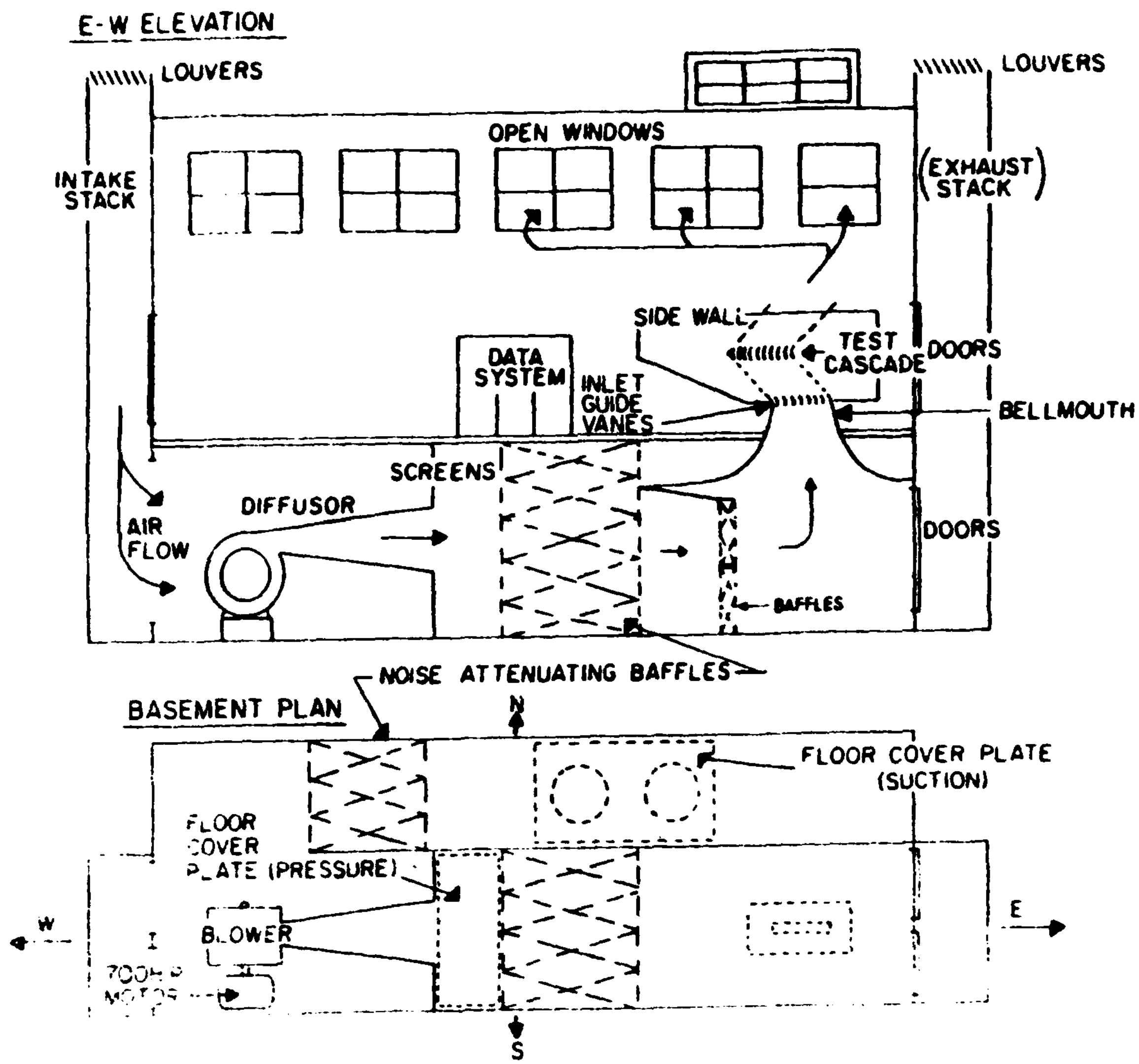


Fig. 1. Subsonic Cascade Facility

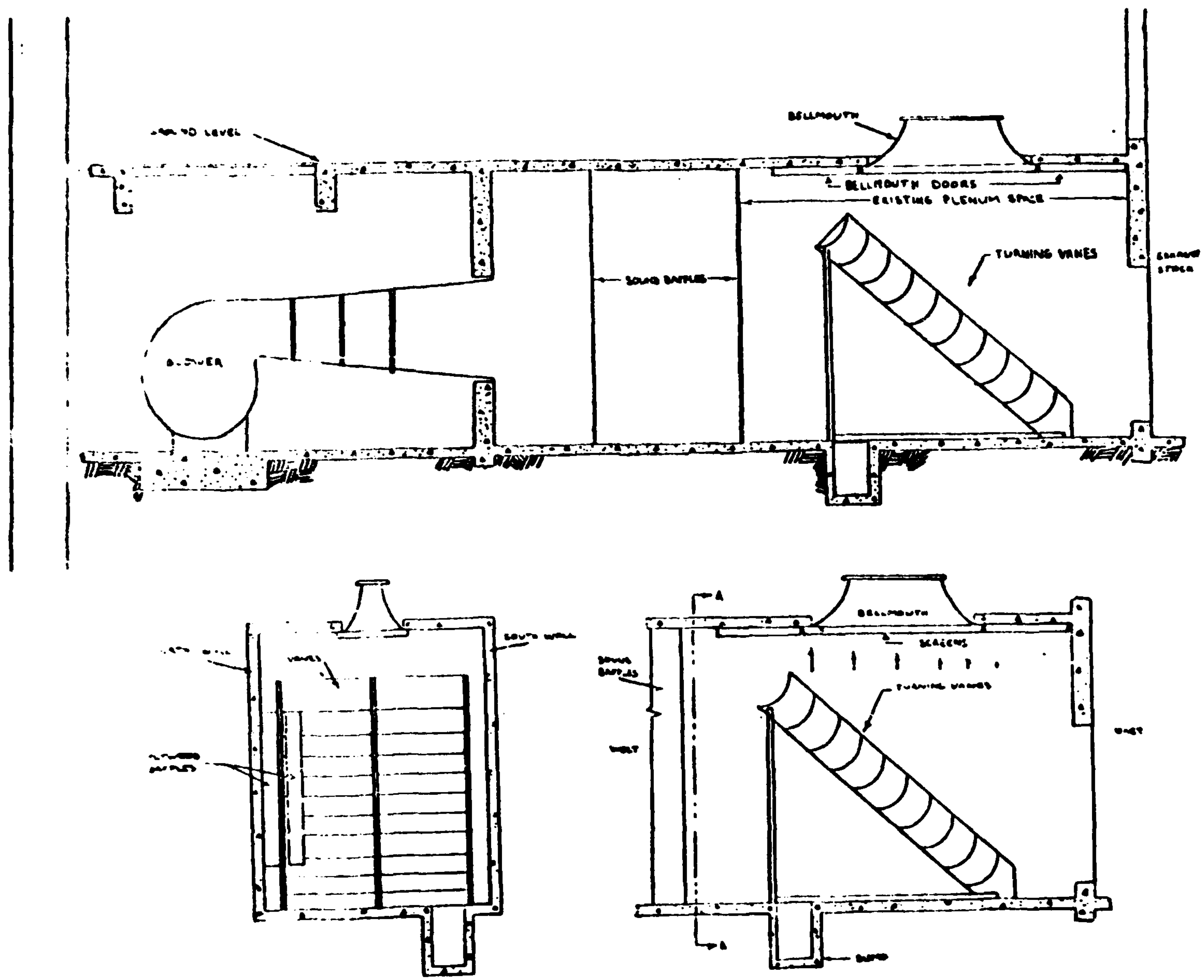


Fig. 2. Plenum Chamber as Modified by Bartocci

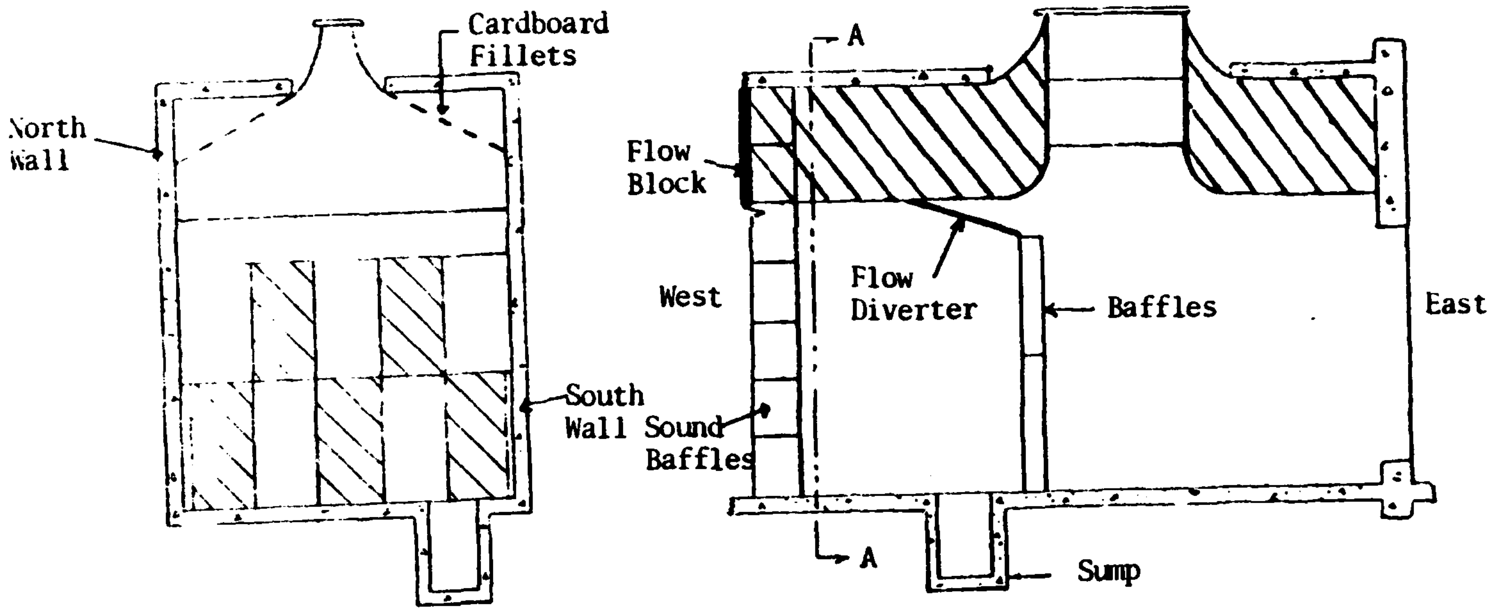


Fig. 3. Plenum Chamber as Modified by Moebius

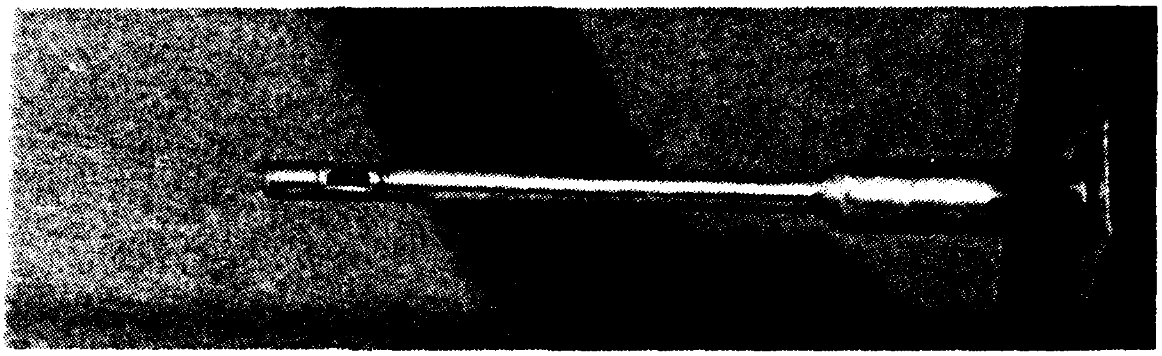


Fig. 4. Lower Plane Survey Probe

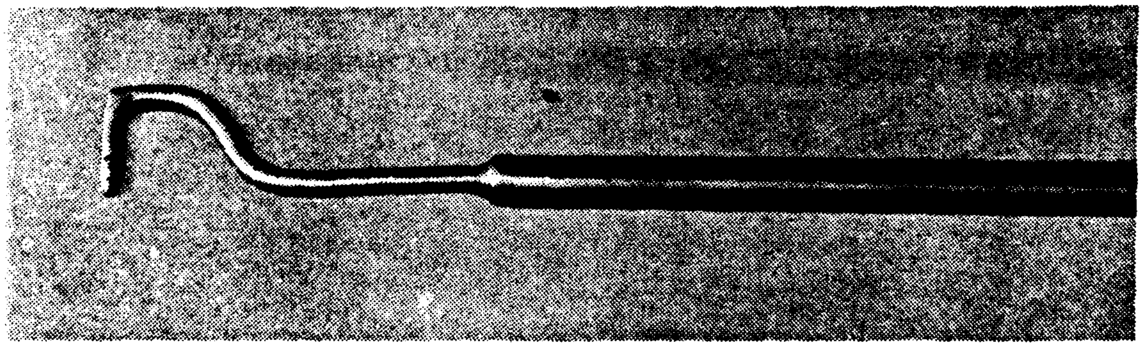


Fig. 5. Upper Plane Survey Probe

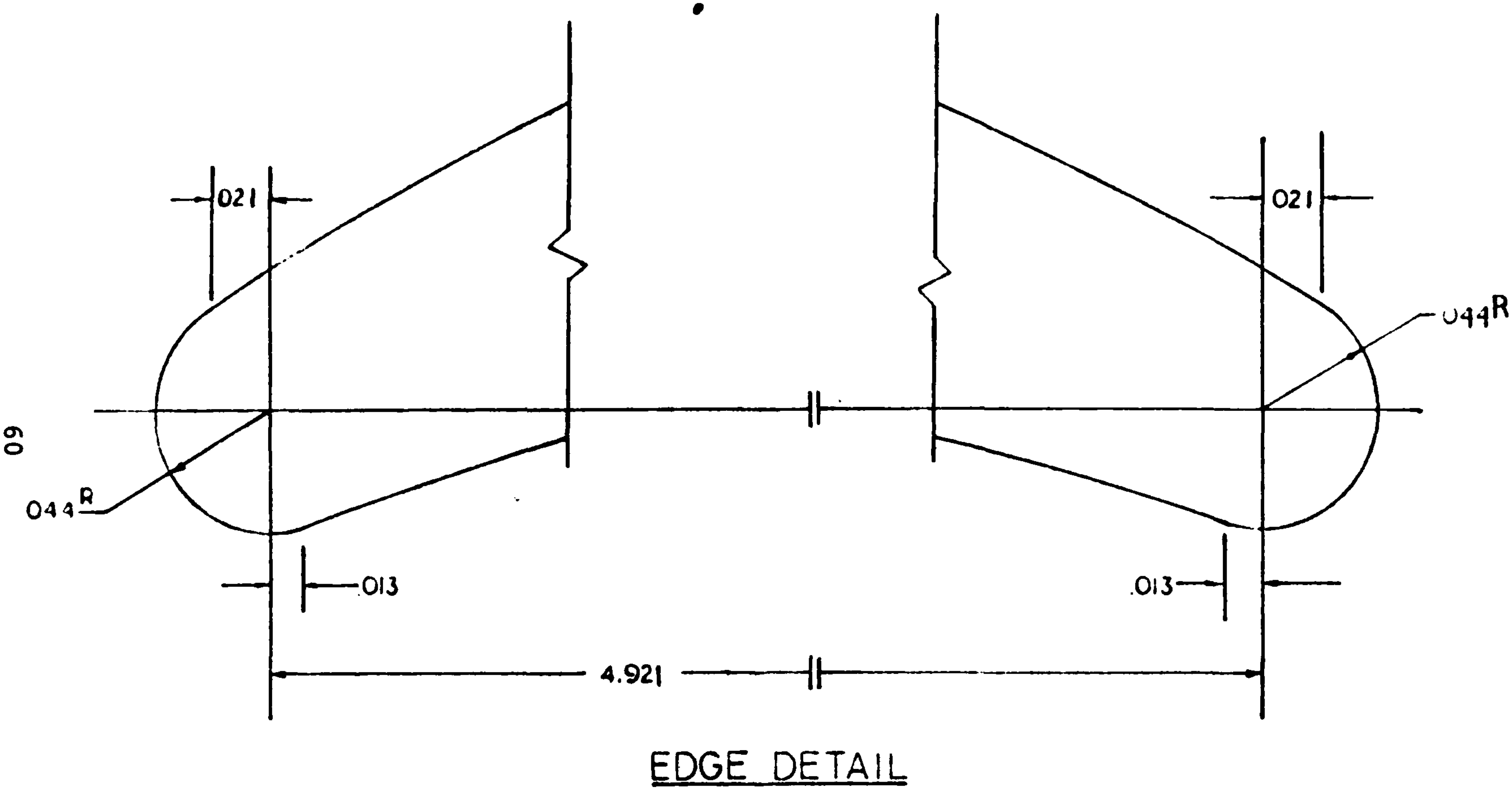


Fig. 6. Blade Edge Detail

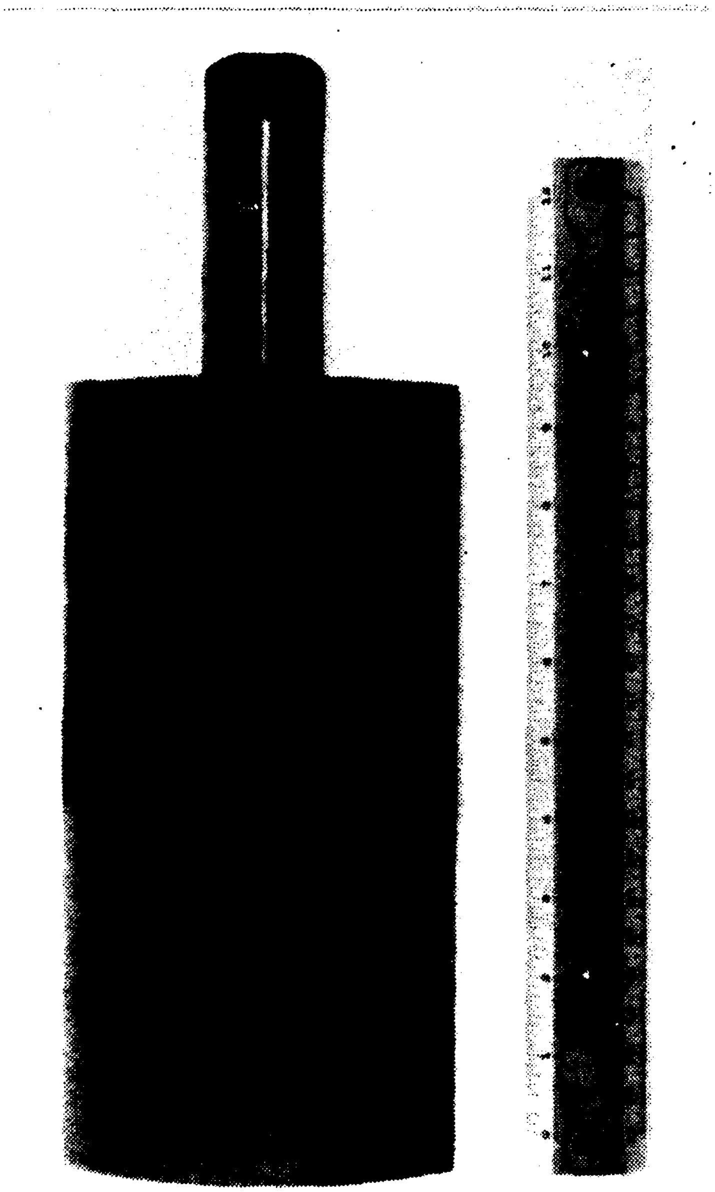


Fig. 7. Photograph of Instrumented Blade

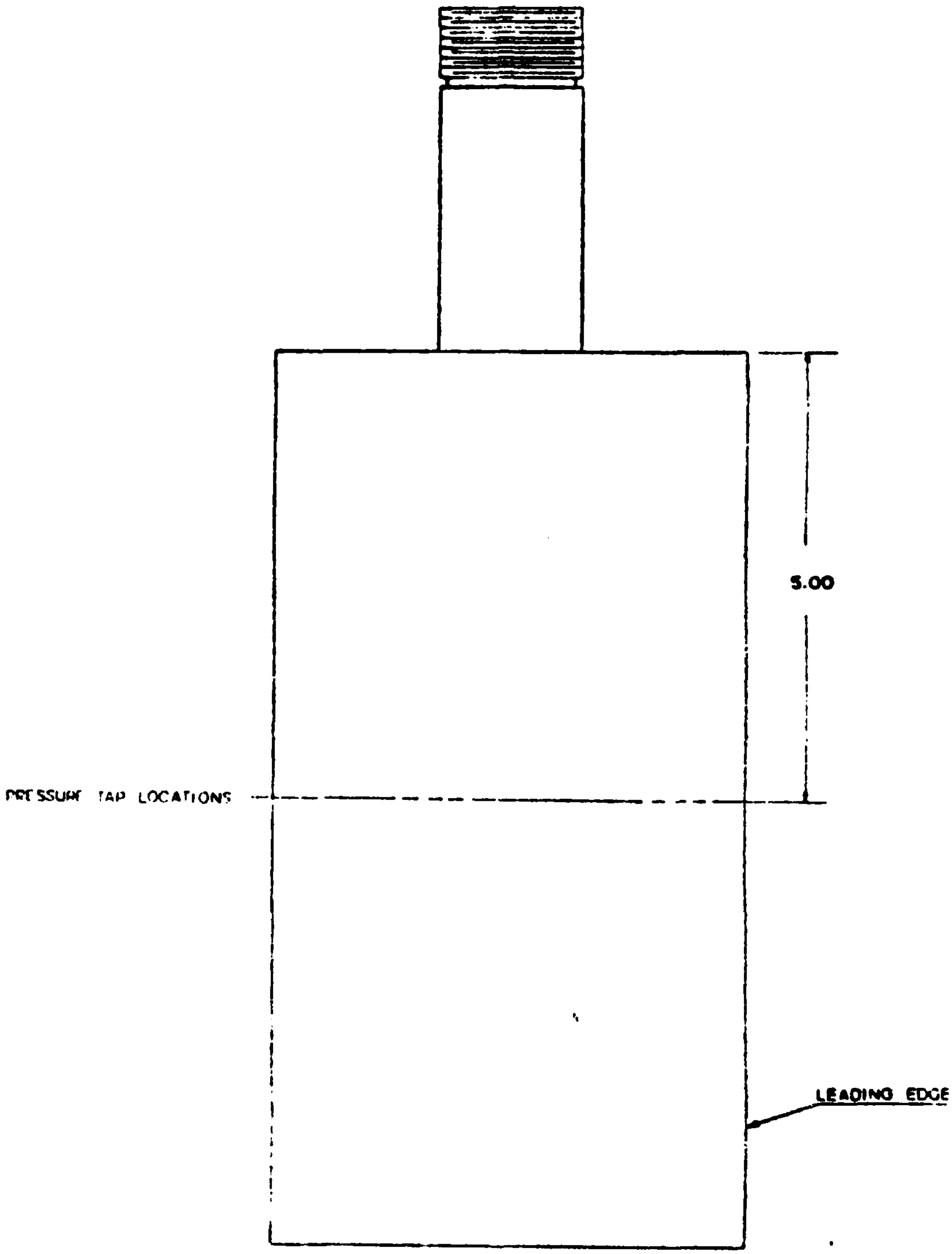


Fig. 8. Instrumented Blade

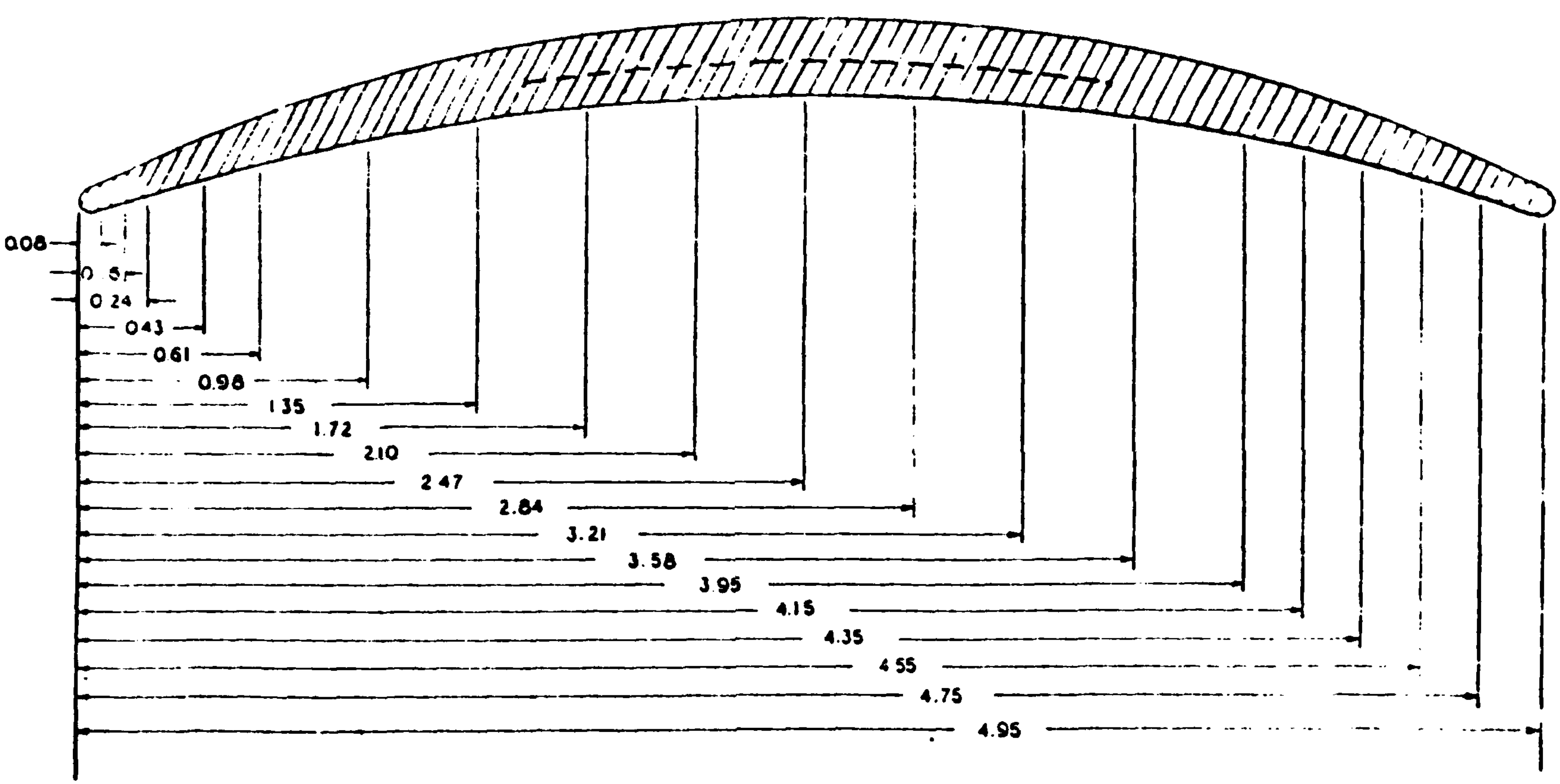


Fig. 9. Instrumented Blade Tap Locations

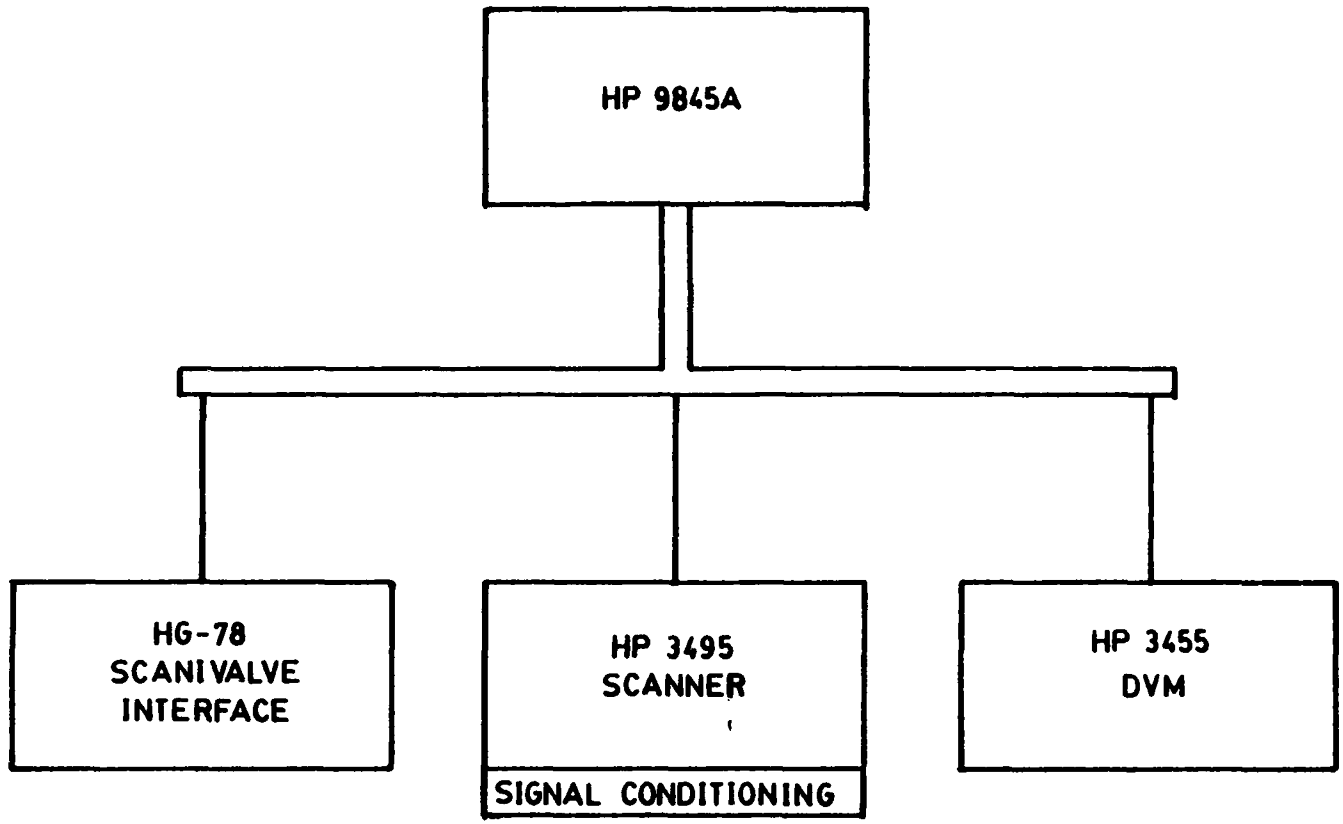


Fig. 10. Data Acquisition System

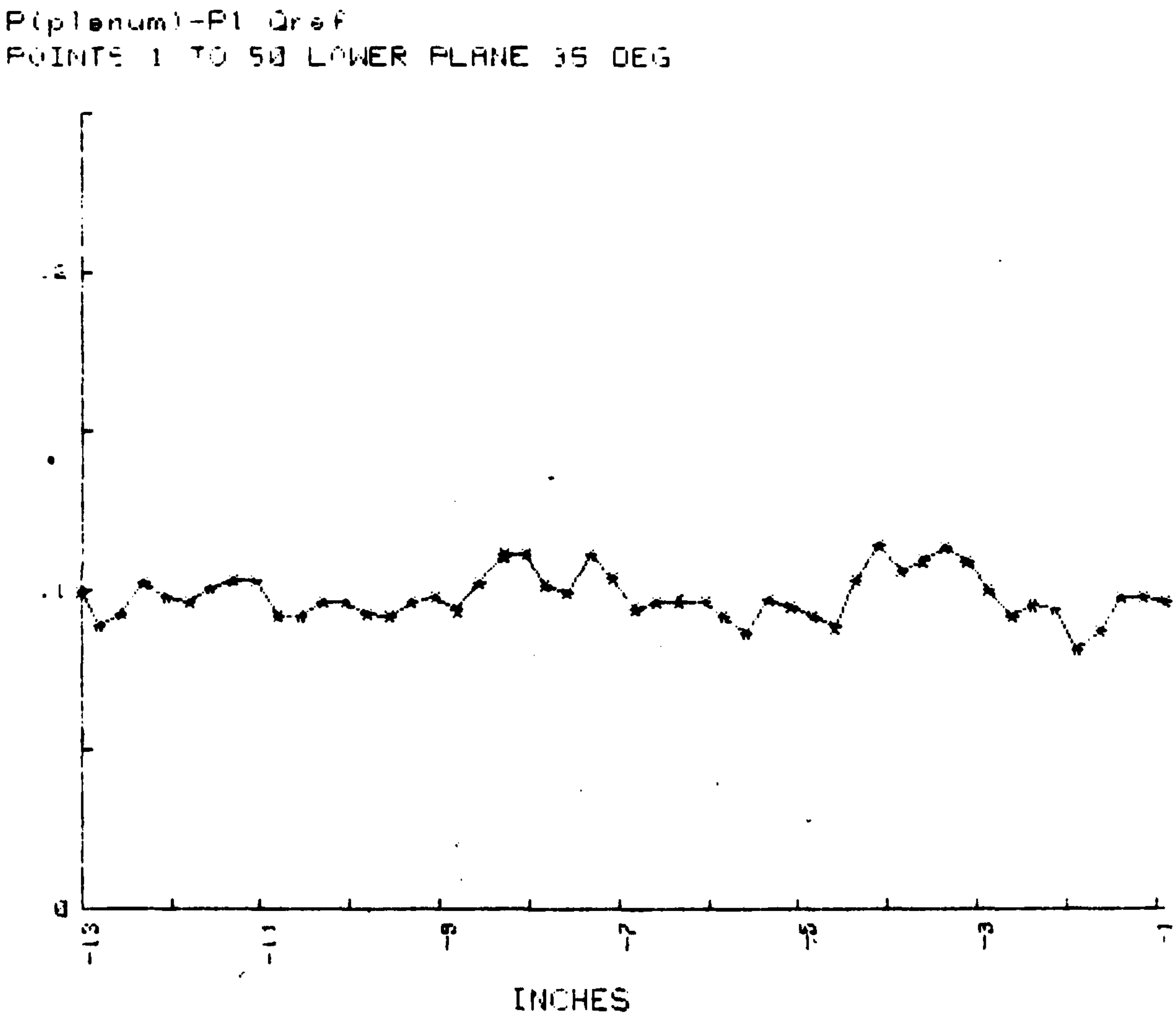


Fig. 11. Probe survey Data at Midspan of Lower Plane End Walls at 35°, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

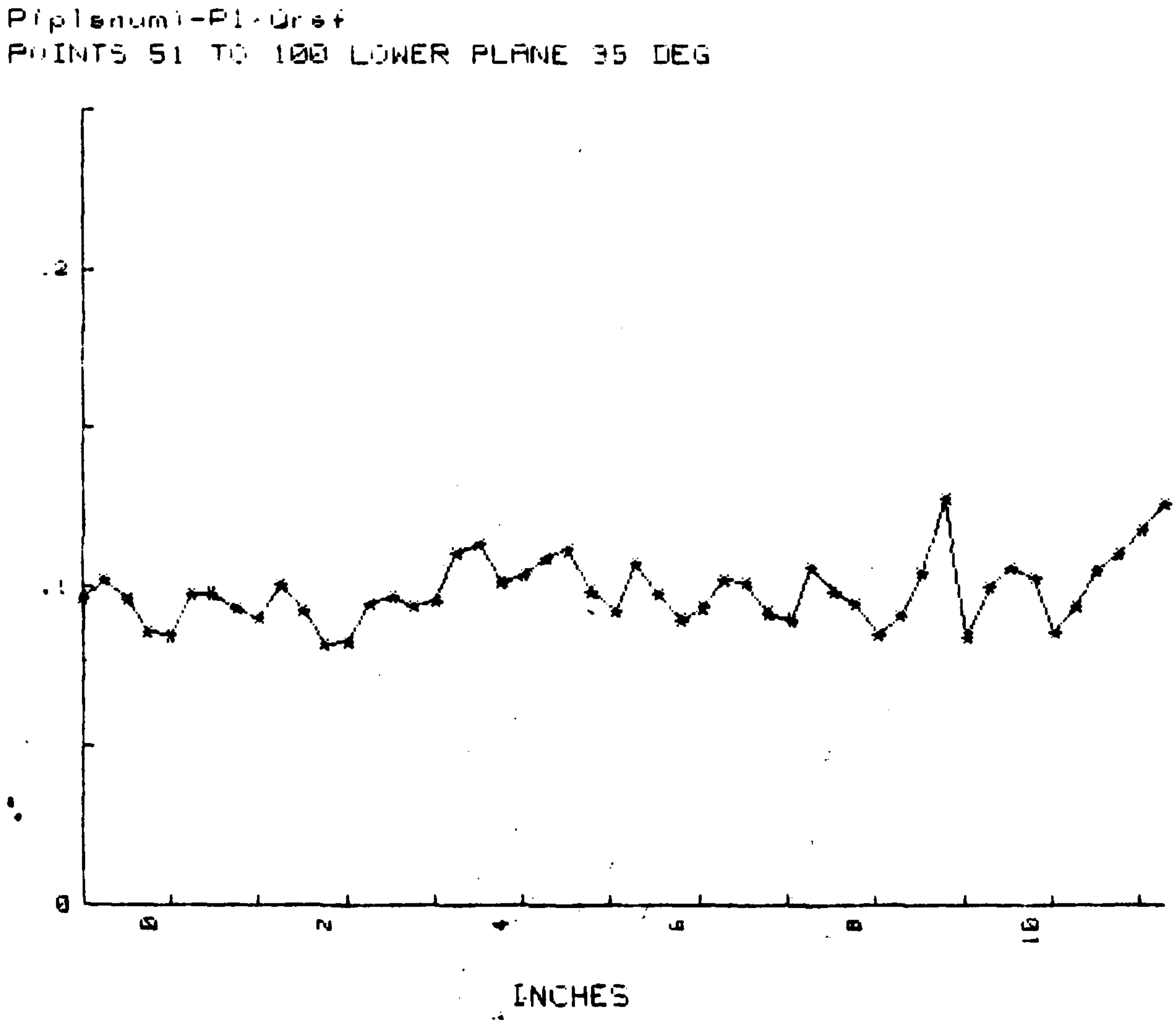


Fig. 12. Probe Survey Data at Midspan of Lower Plane End Walls at 35°, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

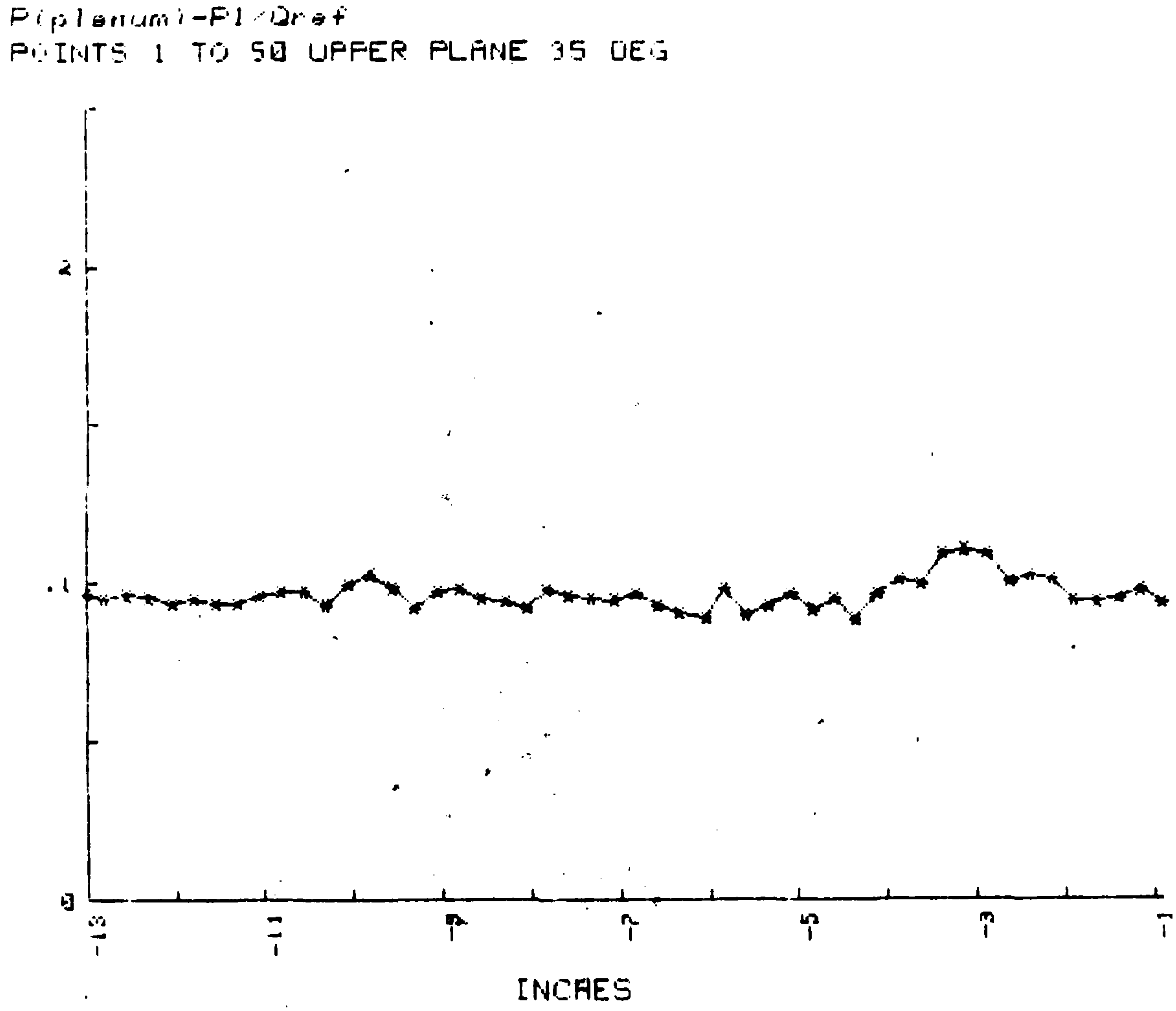


Fig. 13. Probe Survey Data at Midspan of Upper Plane End Walls at 35°, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

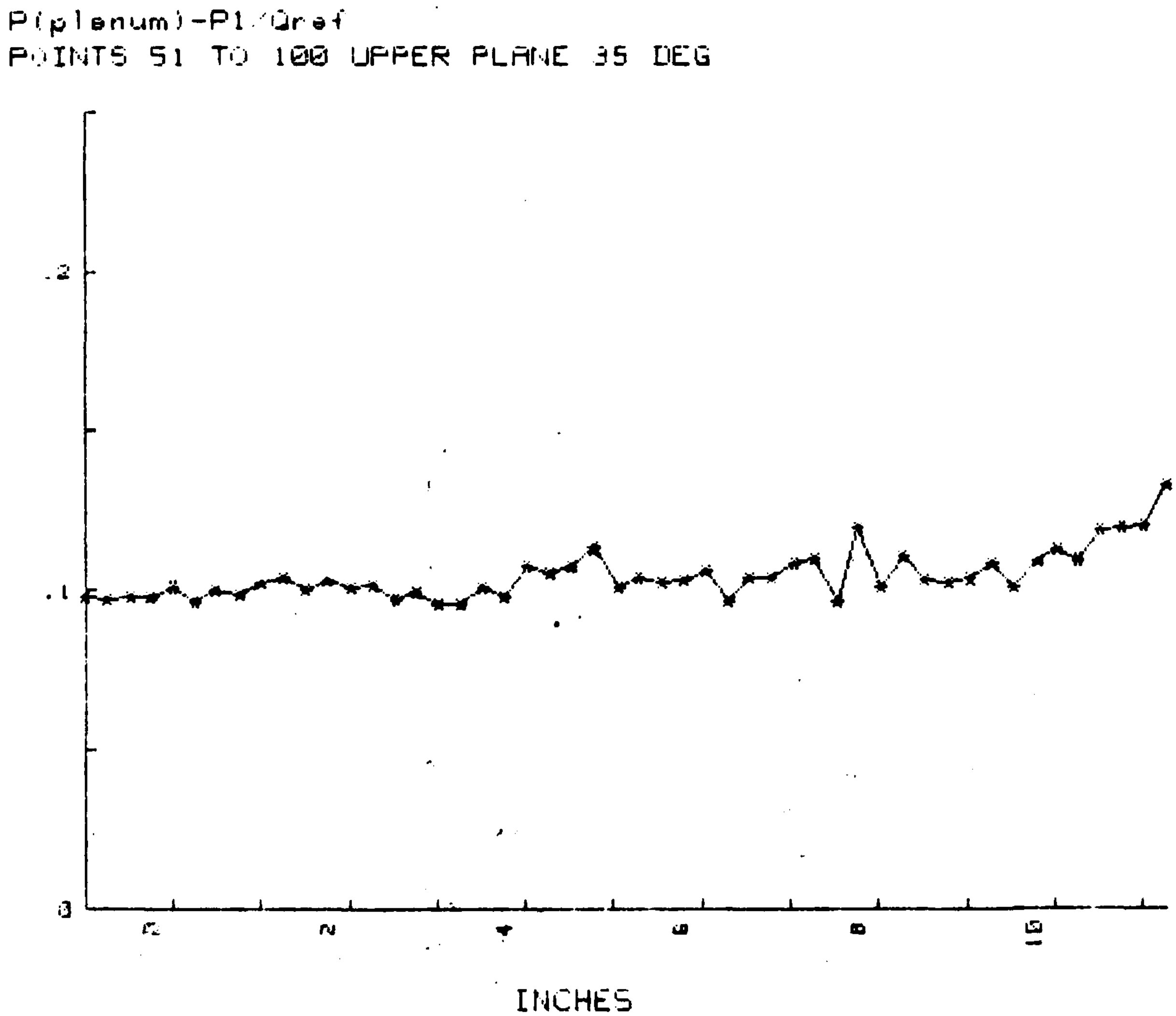


Fig. 14. Probe Survey Data at Midspan of Upper Plane End Walls at 35°, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

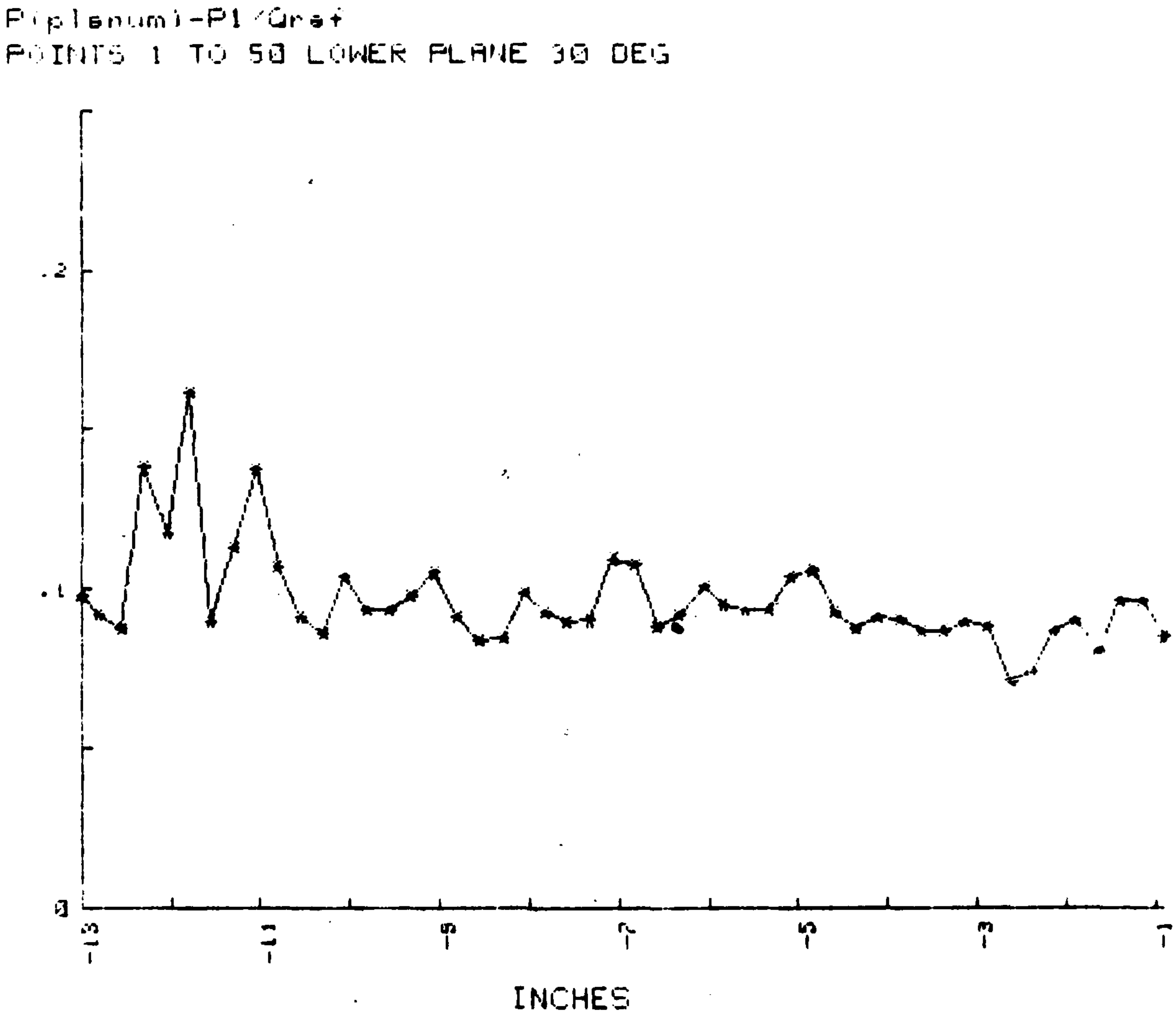


Fig. 15. Probe Survey Data at Midspan of Lower Plane End Walls at 30°, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

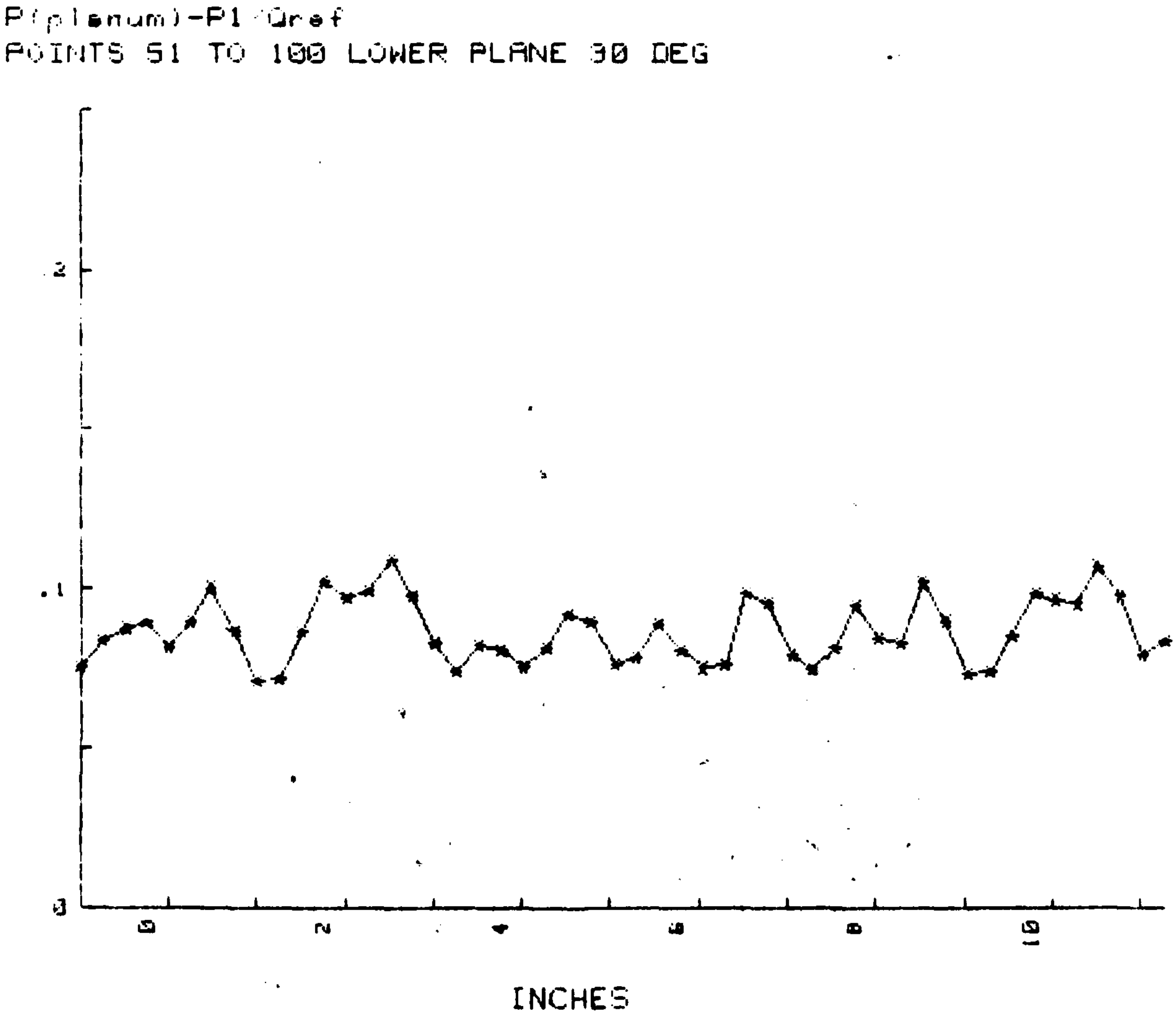


Fig. 16. Probe Survey Data at Midspan of Lower Plane End Walls at 30°, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

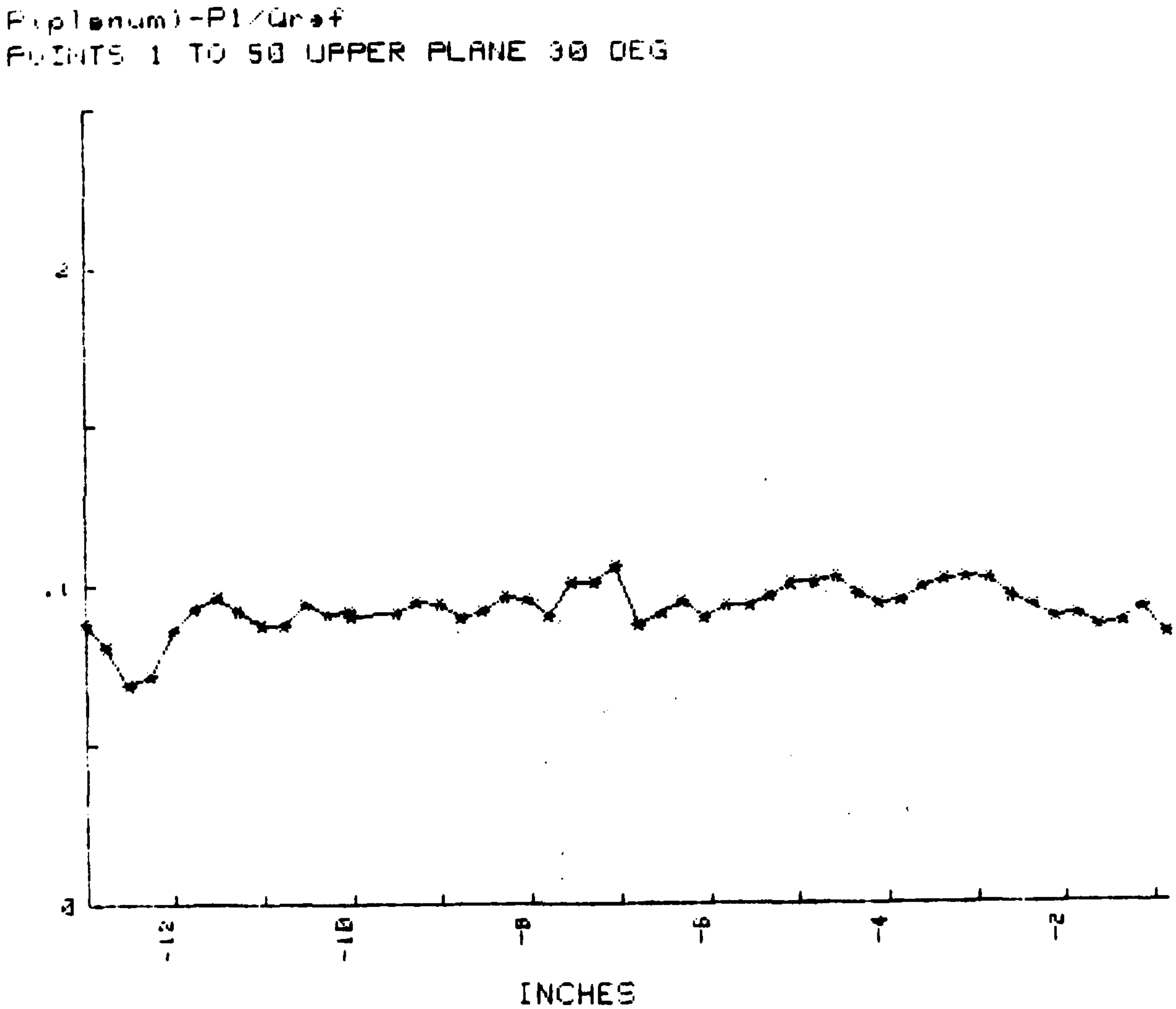


Fig. 17. Probe Survey Data at Midspan of Upper Plane End Walls at 30°, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

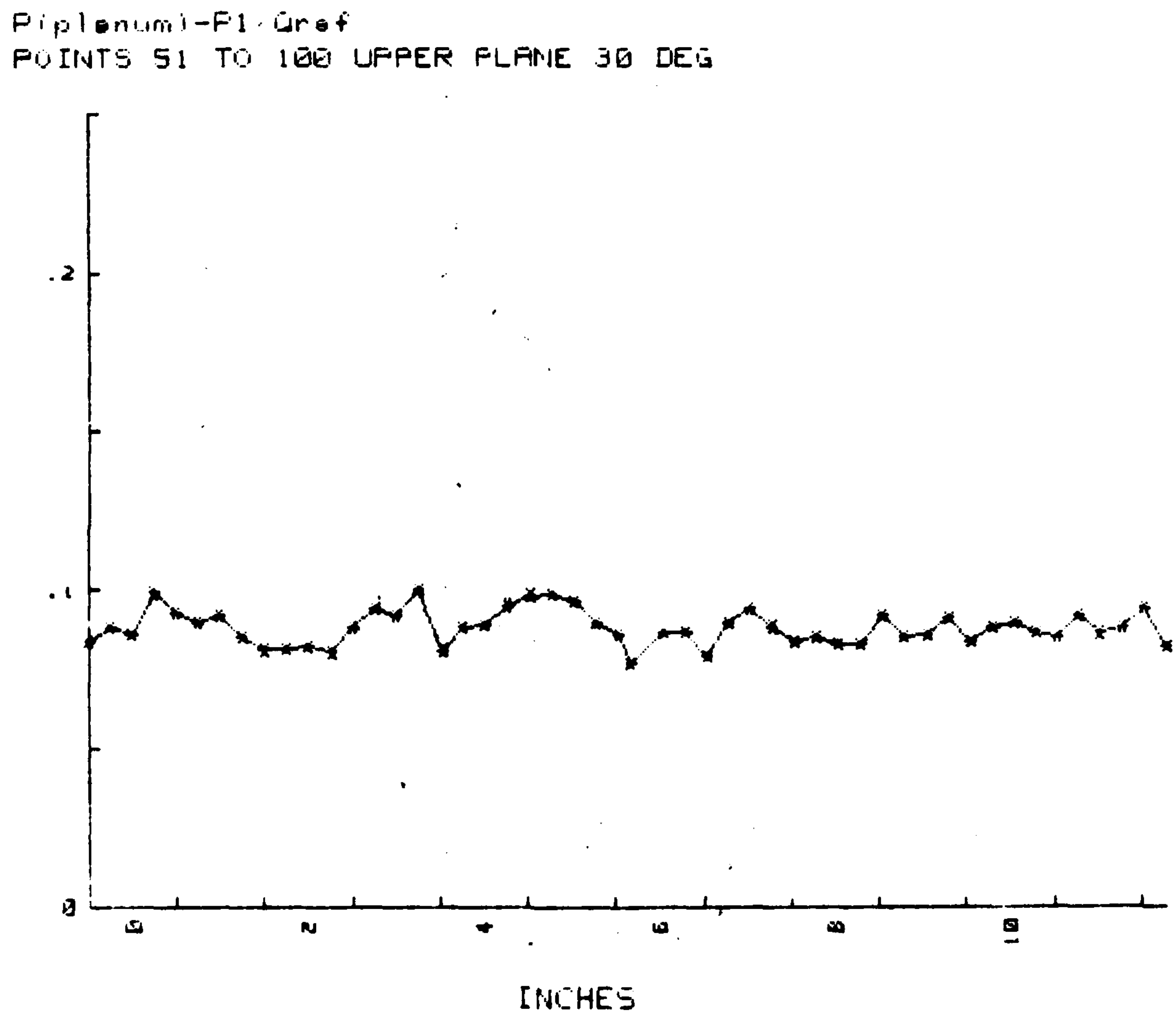


Fig. 18. Probe Survey Data at Midspan of Upper Plane End Walls at 30°, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

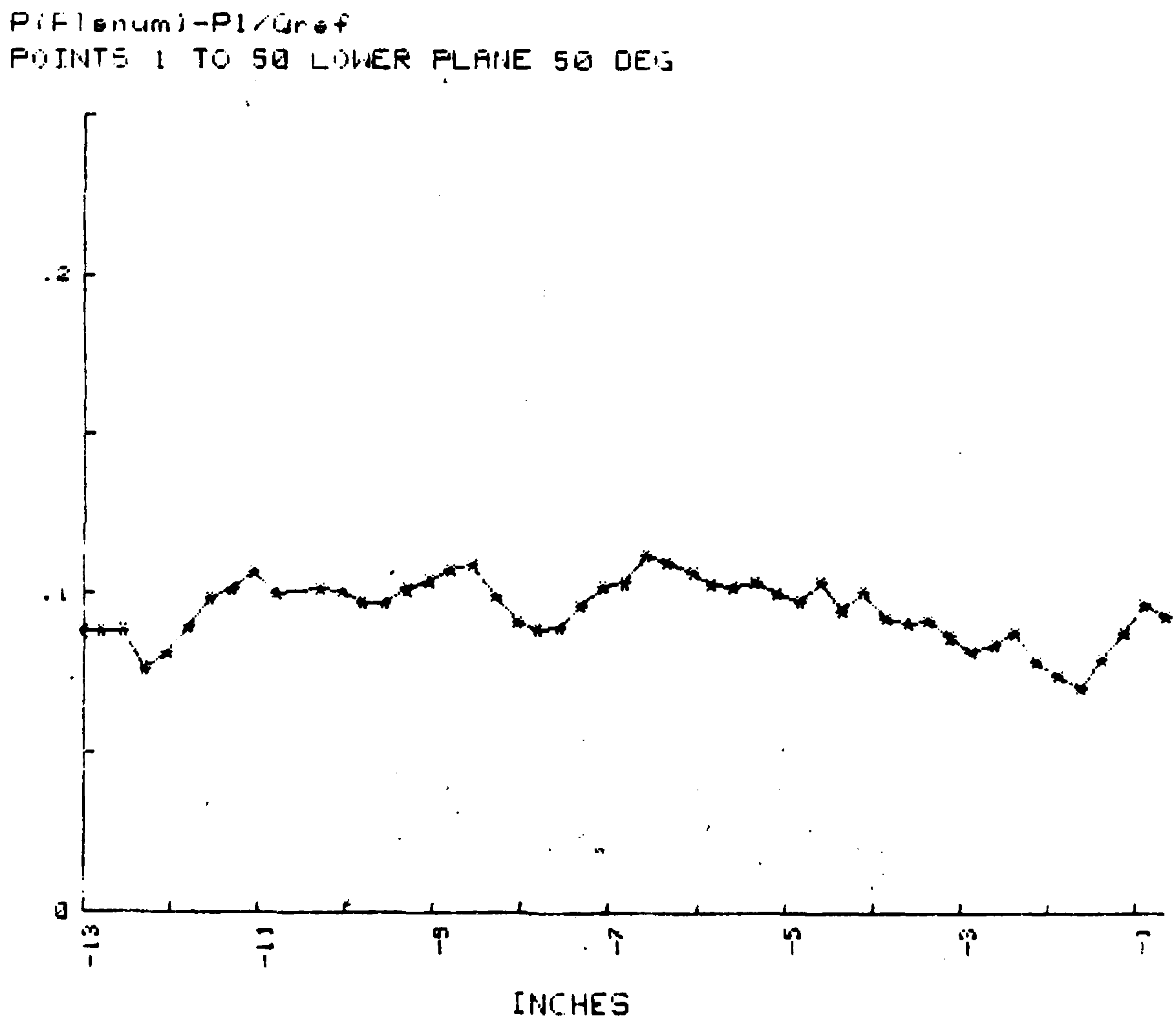


Fig. 19. Probe Survey Data at Midspan of Lower Plane End Walls at 50°, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

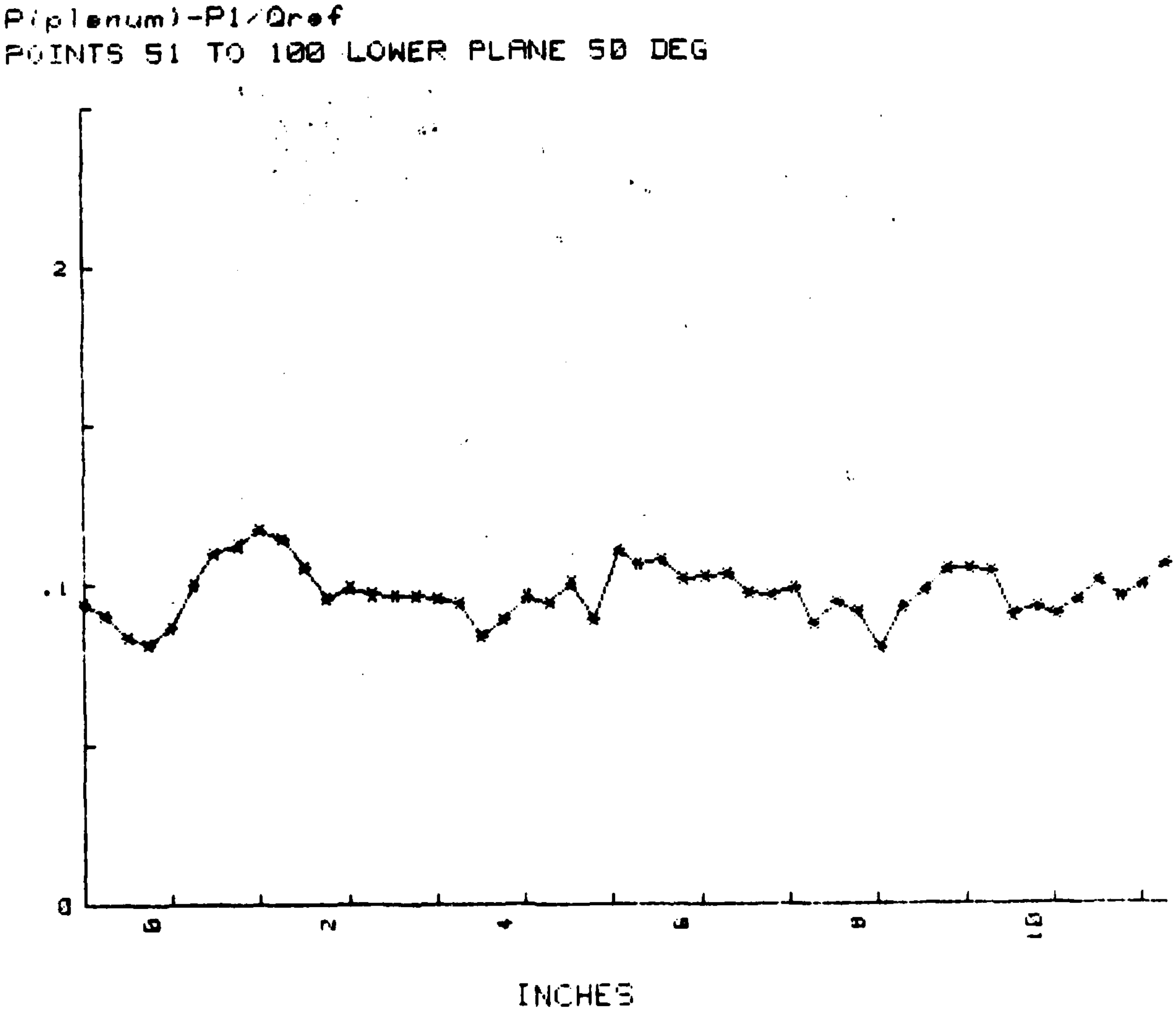


Fig. 20. Probe Survey Data at Midspan of Lower Plane End Walls at 50°, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

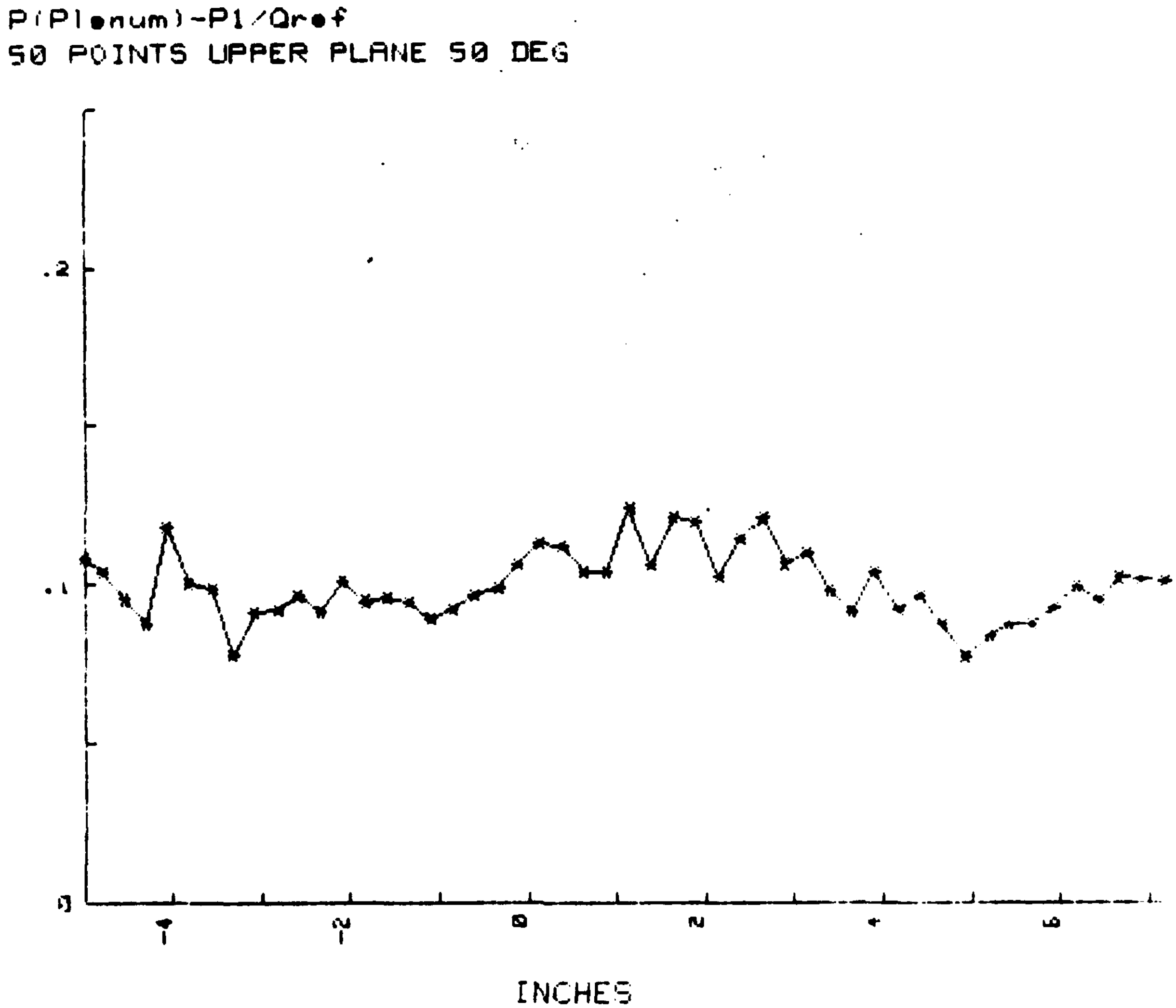


Fig. 21. Probe Survey Data at Midspan of Upper Plane End Walls at 50°
$(P_{PLENUM} - P_t)/Q_{ref}$

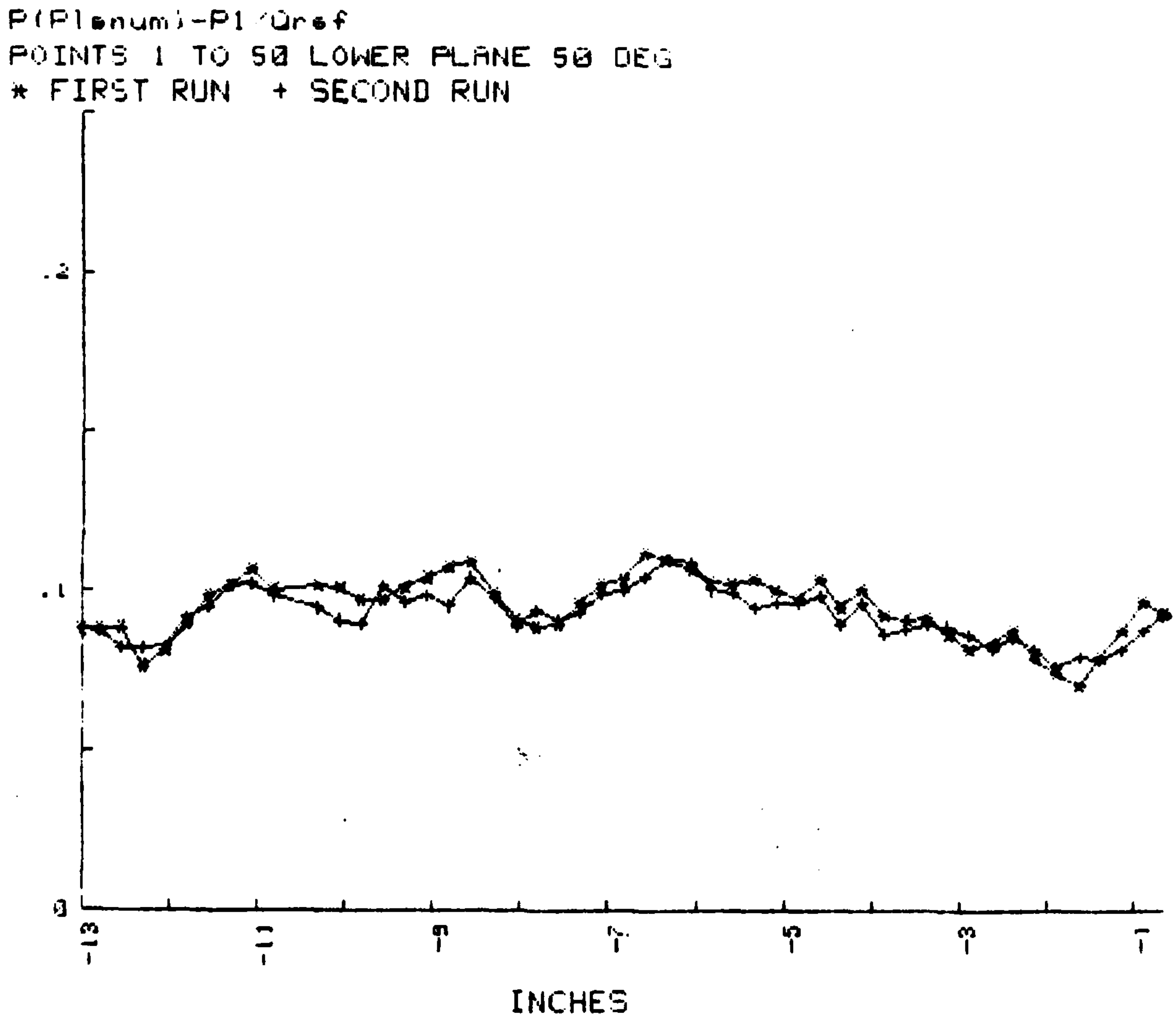


Fig. 22. Probe Survey Data at Midspan at Lower Plane End Walls at 50°, Two Runs, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

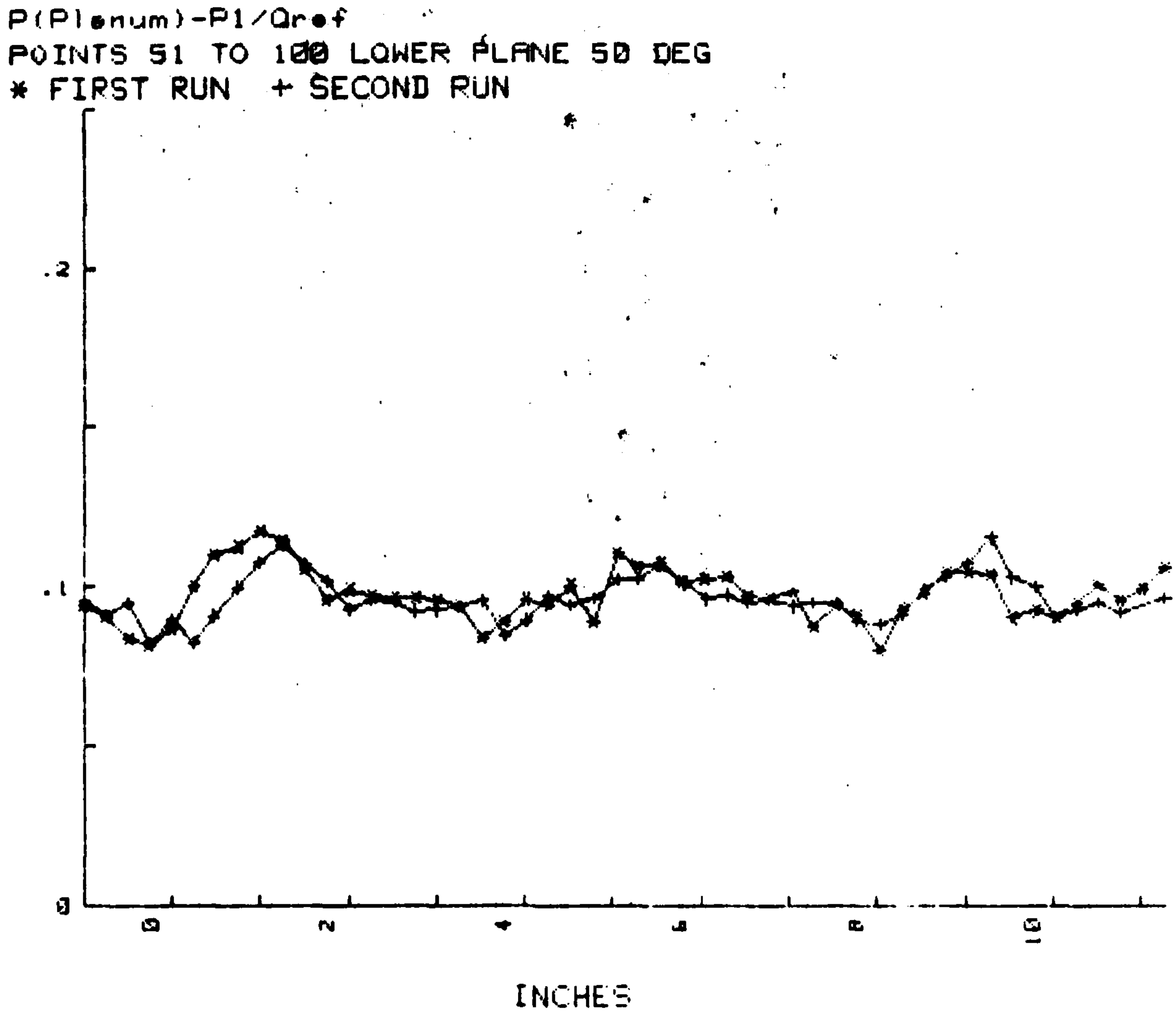


Fig. 23. Probe Survey Data at Midspan at Lower Plane End Walls at 50°, Two Runs, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

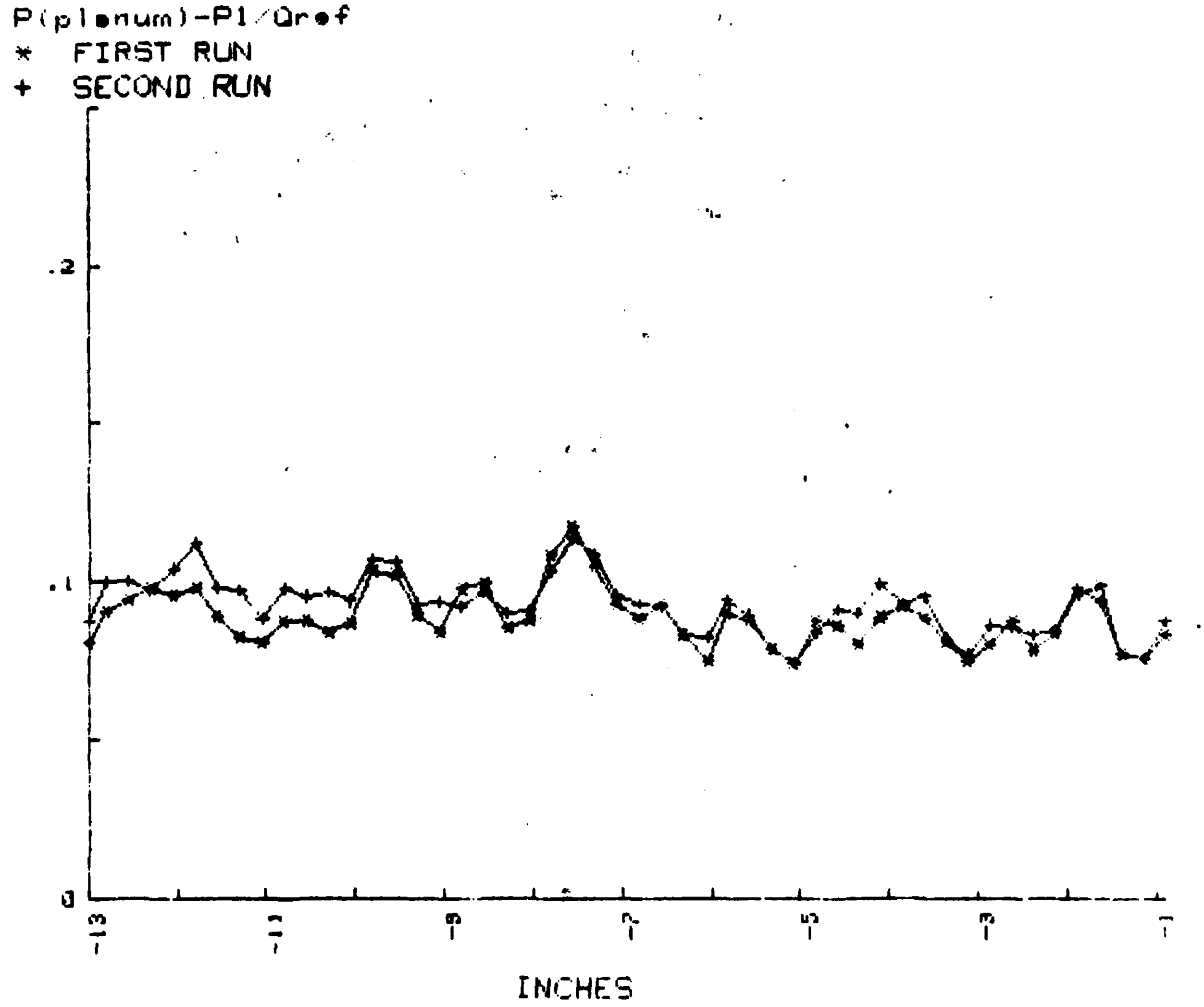


Fig. 24. Probe Survey Data at Midspan at Lower Plane End Walls at 30°, Two Runs, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

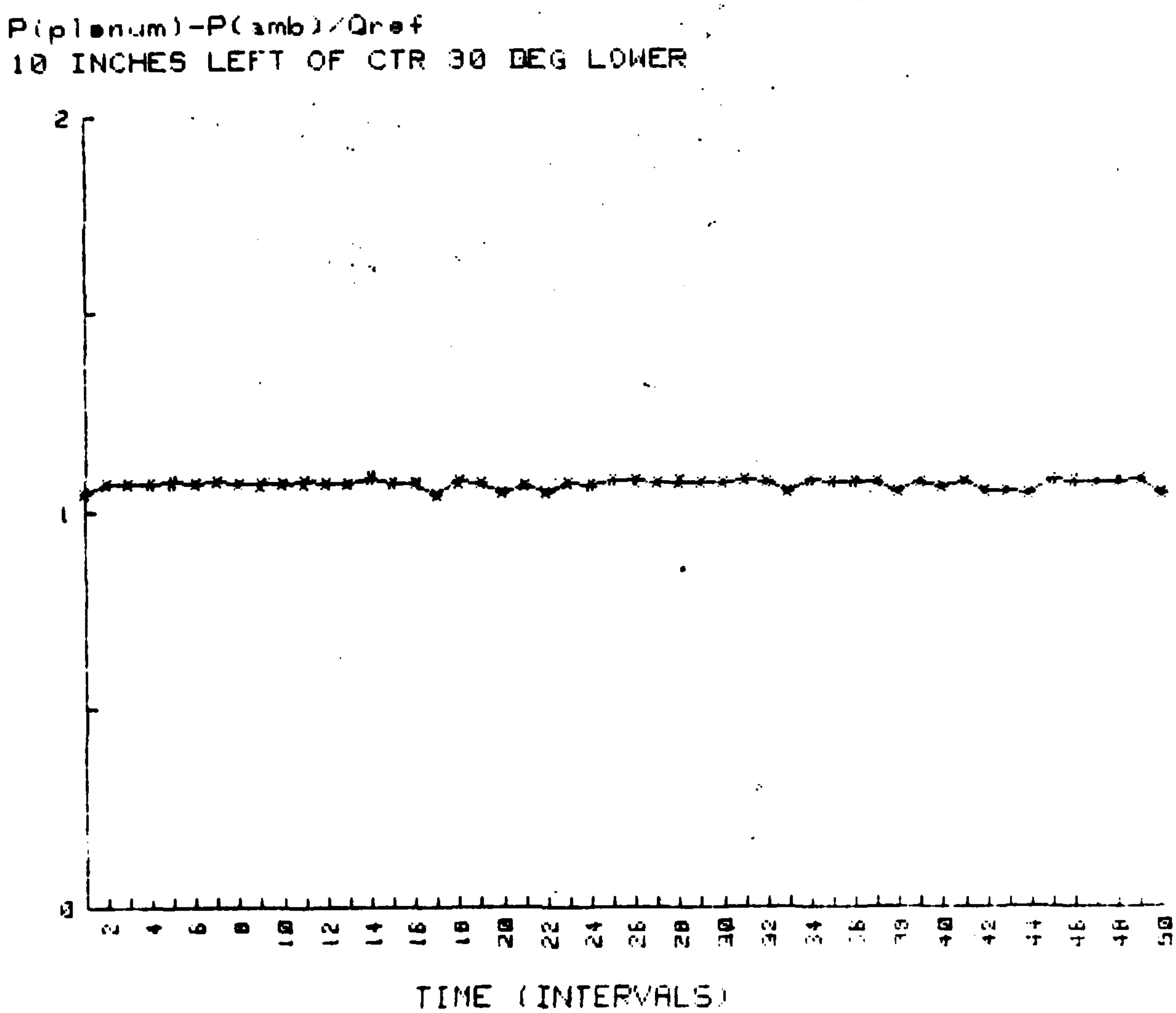


Fig. 25. Repetitive Samples with Fixed Probe Position (10" Left of CTR Midspan, End Walls 30°, Lower Plane $(P_{PLENUM} - P_{AMB})/Q_{ref}$)

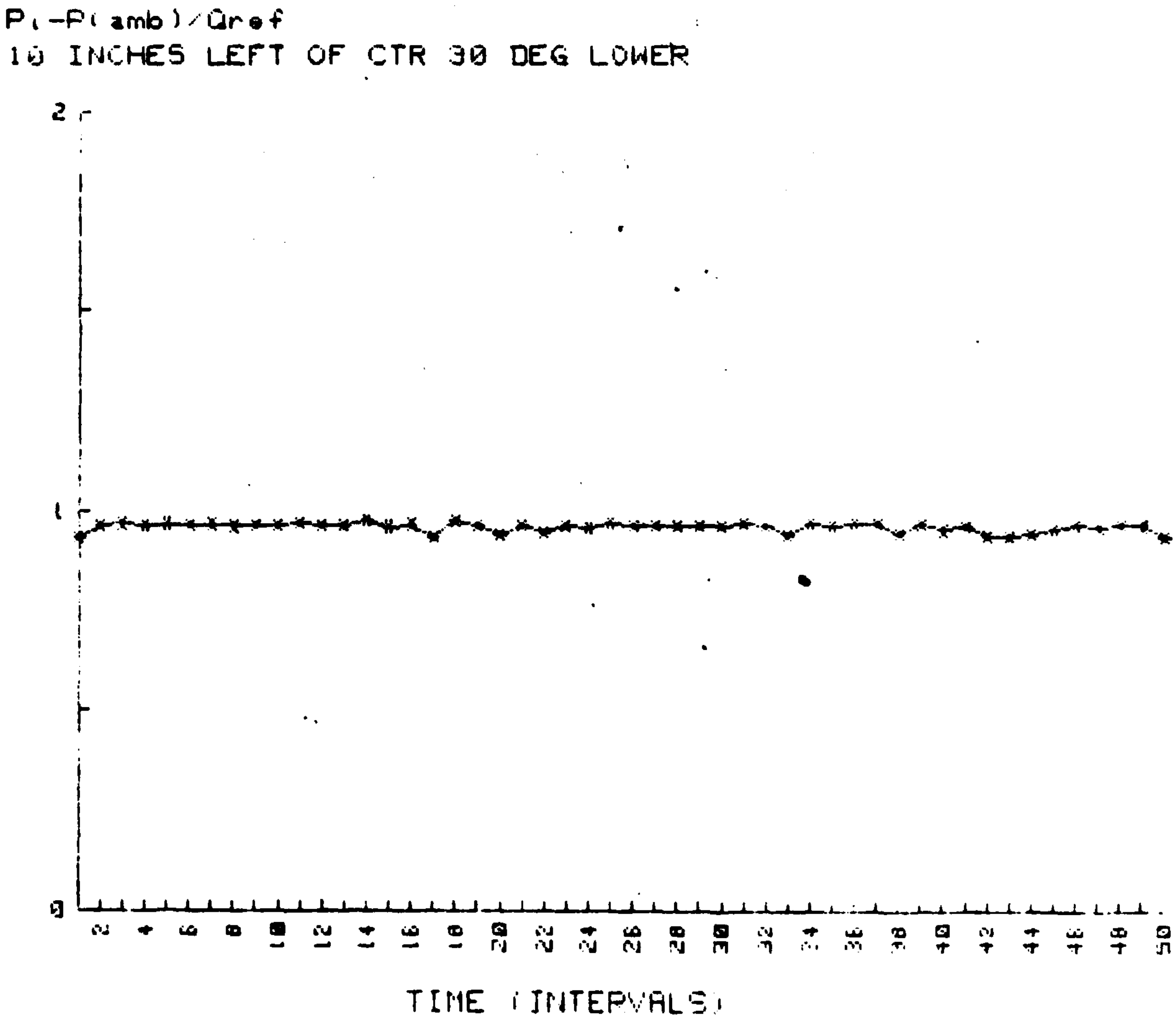


Fig. 26. Repetitive Samples with Fixed Probe Position (10" Left of CTR Midspan, End Walls at 30°, Lower Plane $(P_t - P_{AMB})/Q_{ref}$)

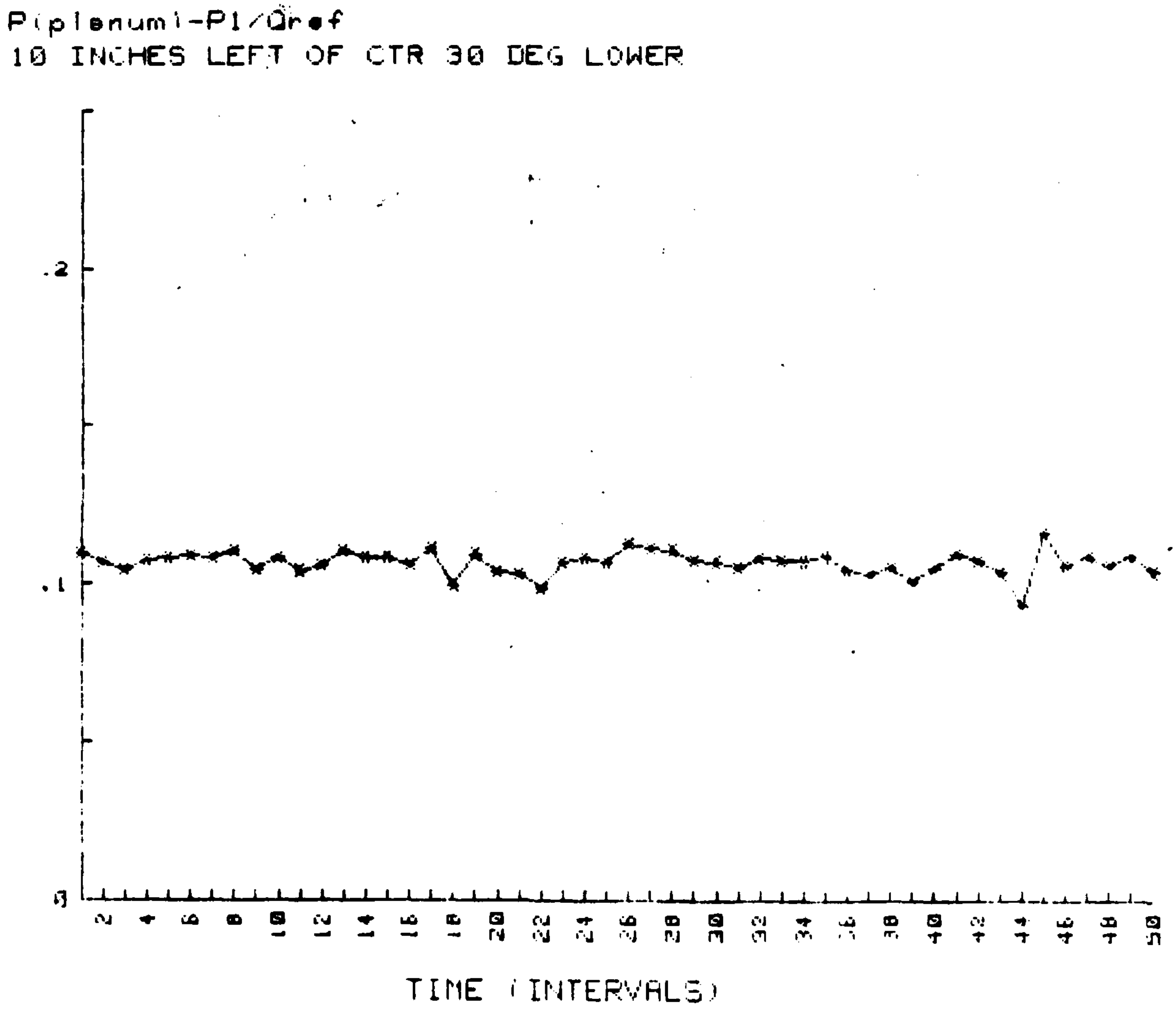


Fig. 27. Repetitive Samples with Fixed Probe Position (10" Left of CTR Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_t)/Q_{ref}$

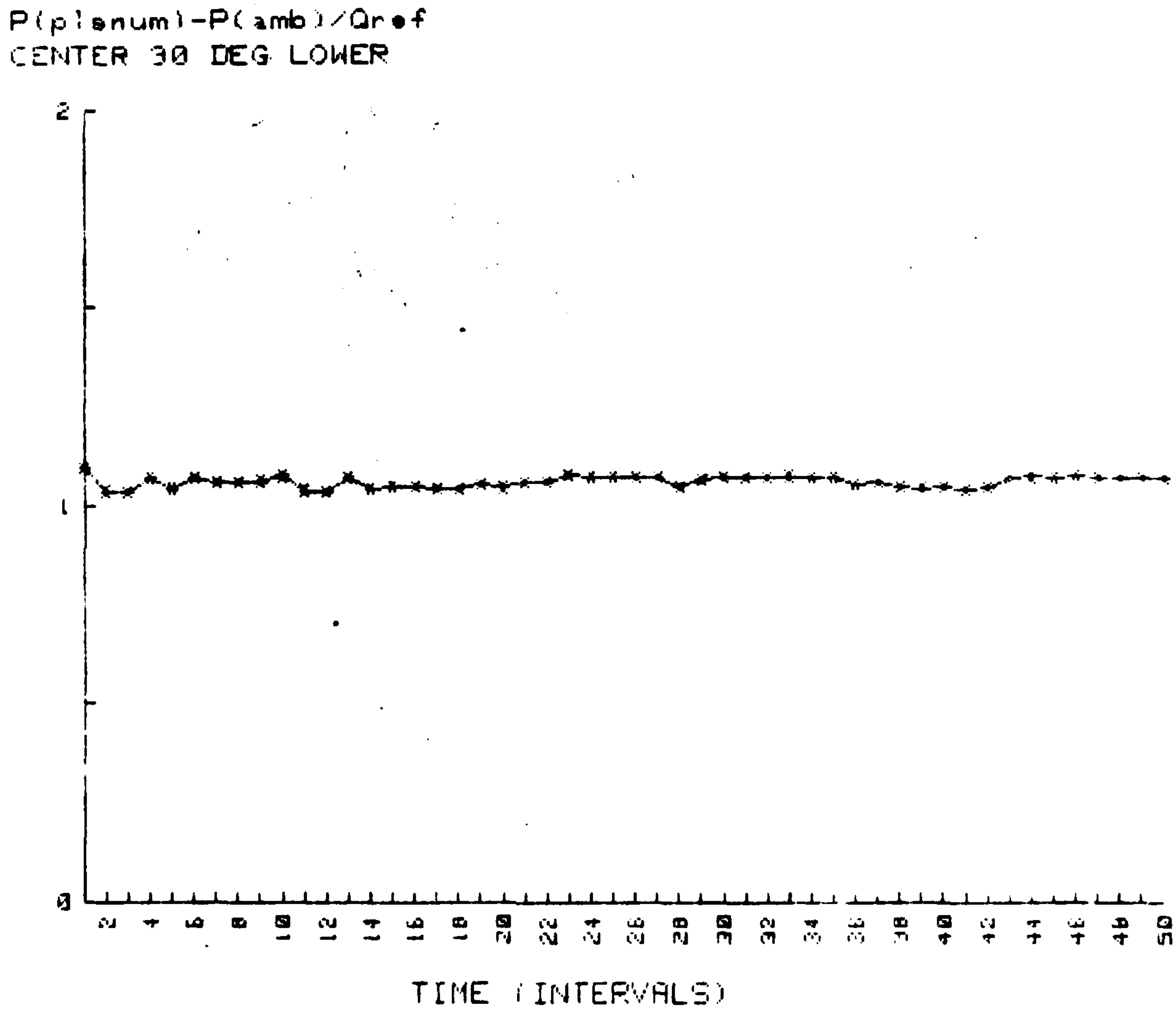


Fig. 28. Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_{AMB})/Q_{ref}$)

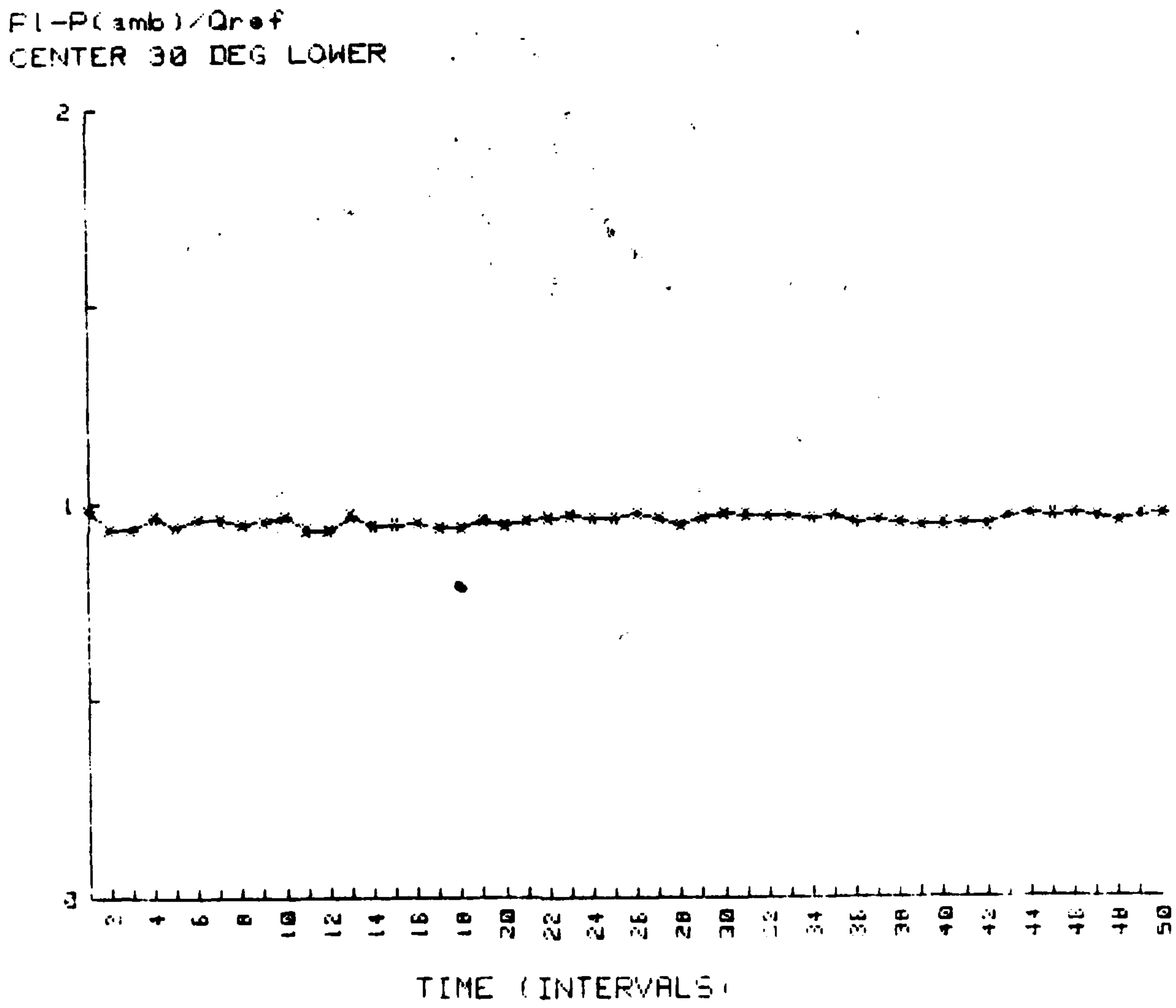


Fig. 29. Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_t - P_{AMB})/Q_{ref}$)

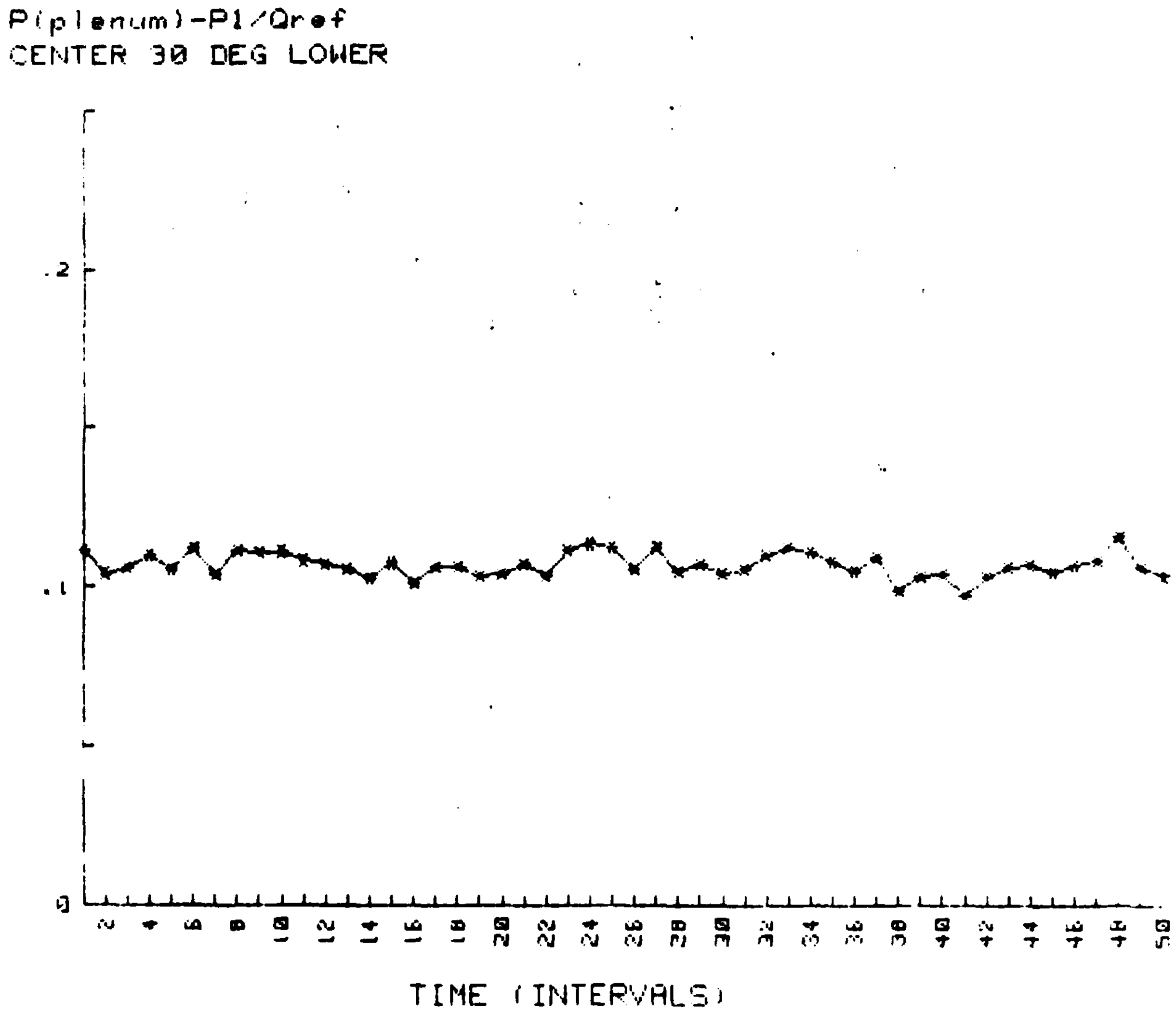


Fig. 30. Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_t)/Q_{ref}$)

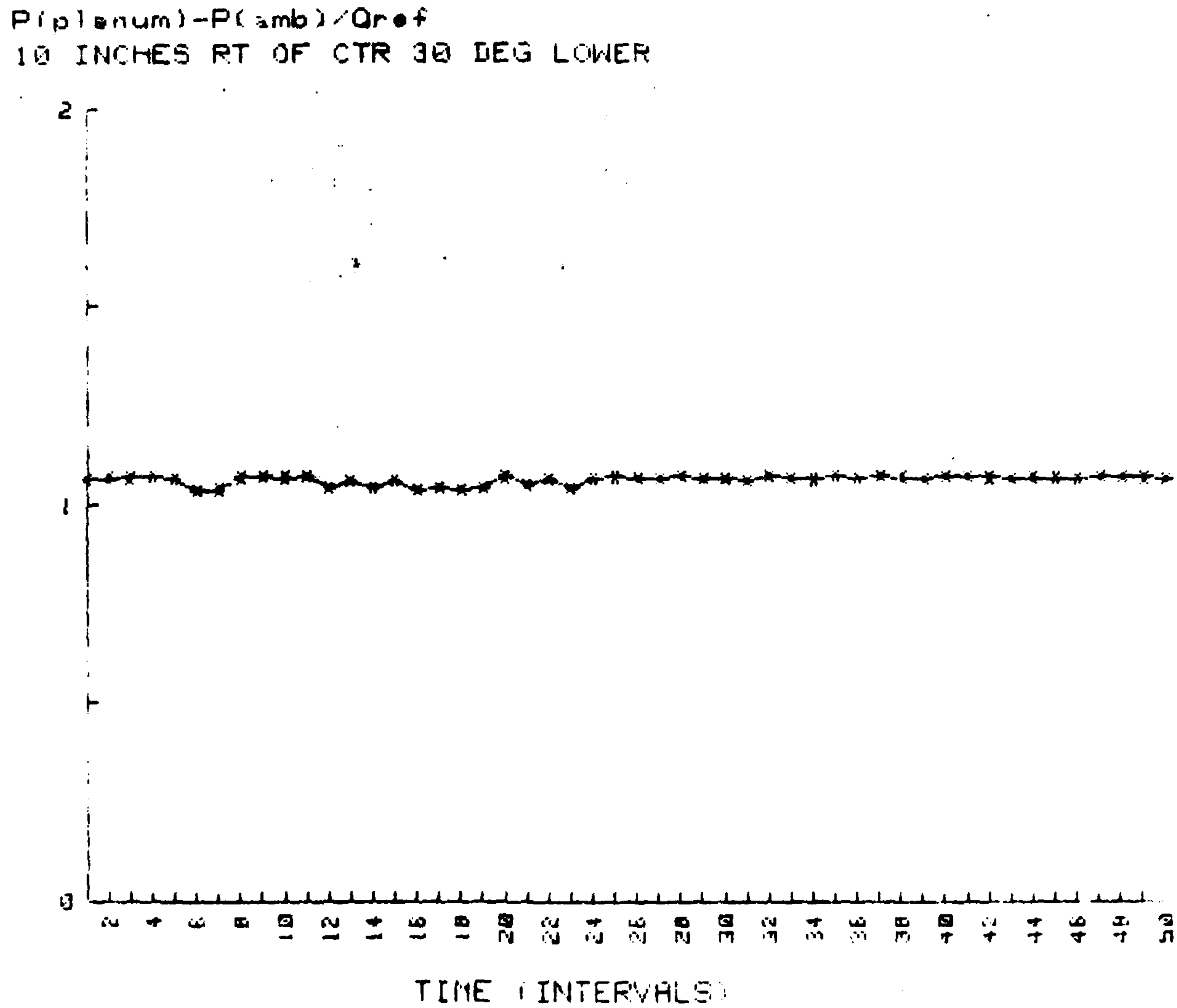


Fig. 31. Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_{AMB})/Q_{ref}$)

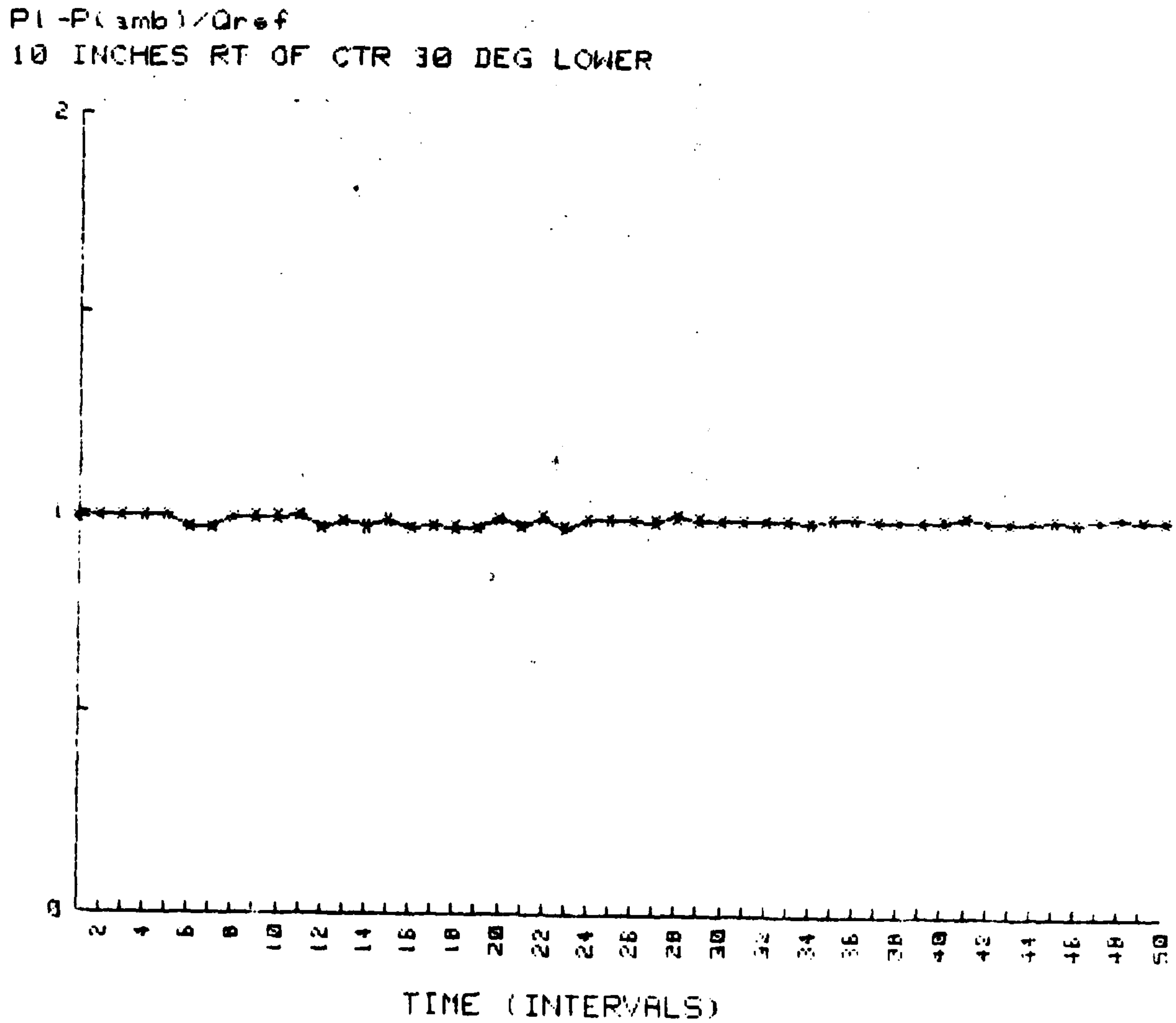


Fig. 32. Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_t - P_{AMB})/Q_{ref}$)

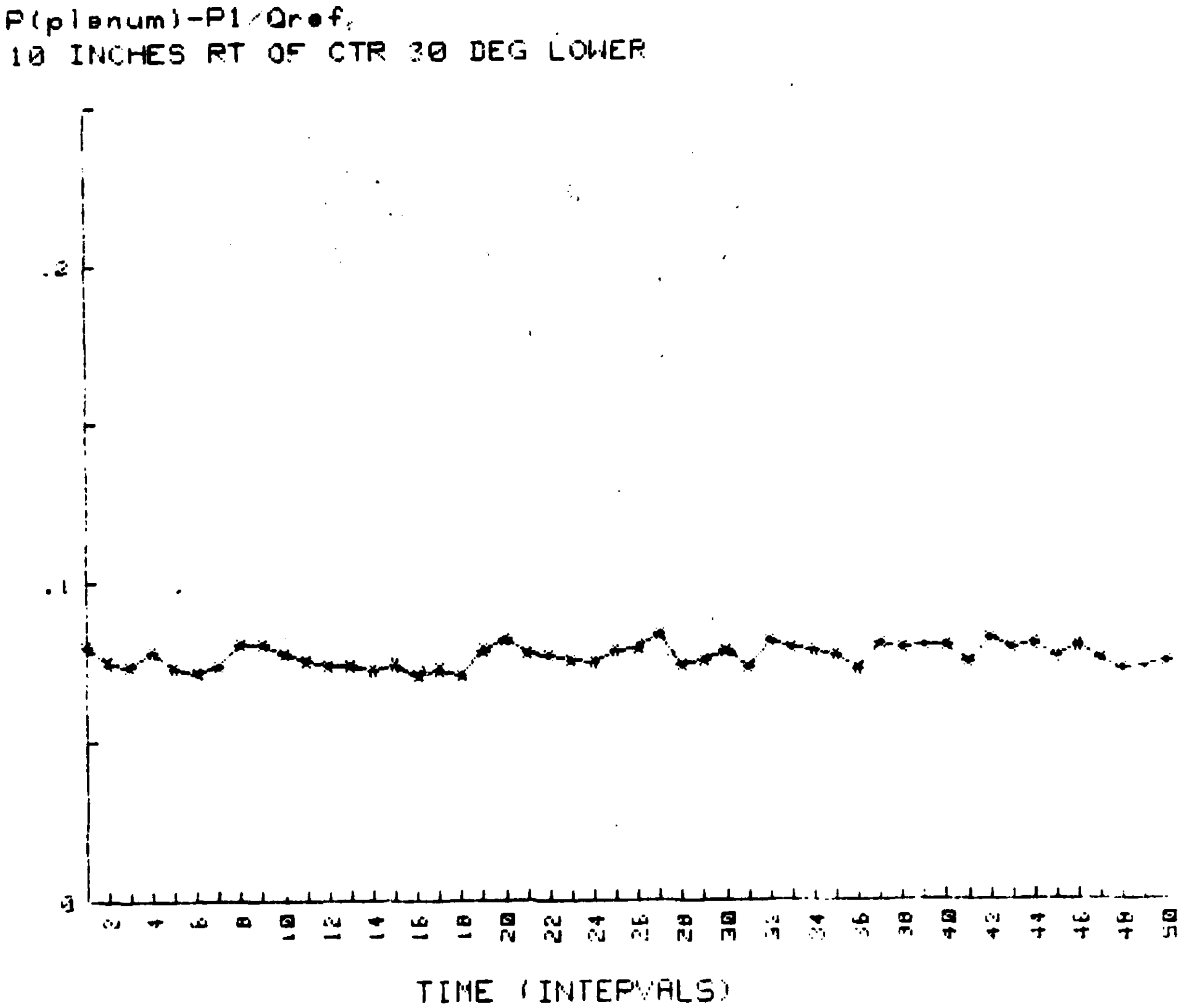


Fig. 33. Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_{PLENUM} - P_t)/Q_{ref}$)

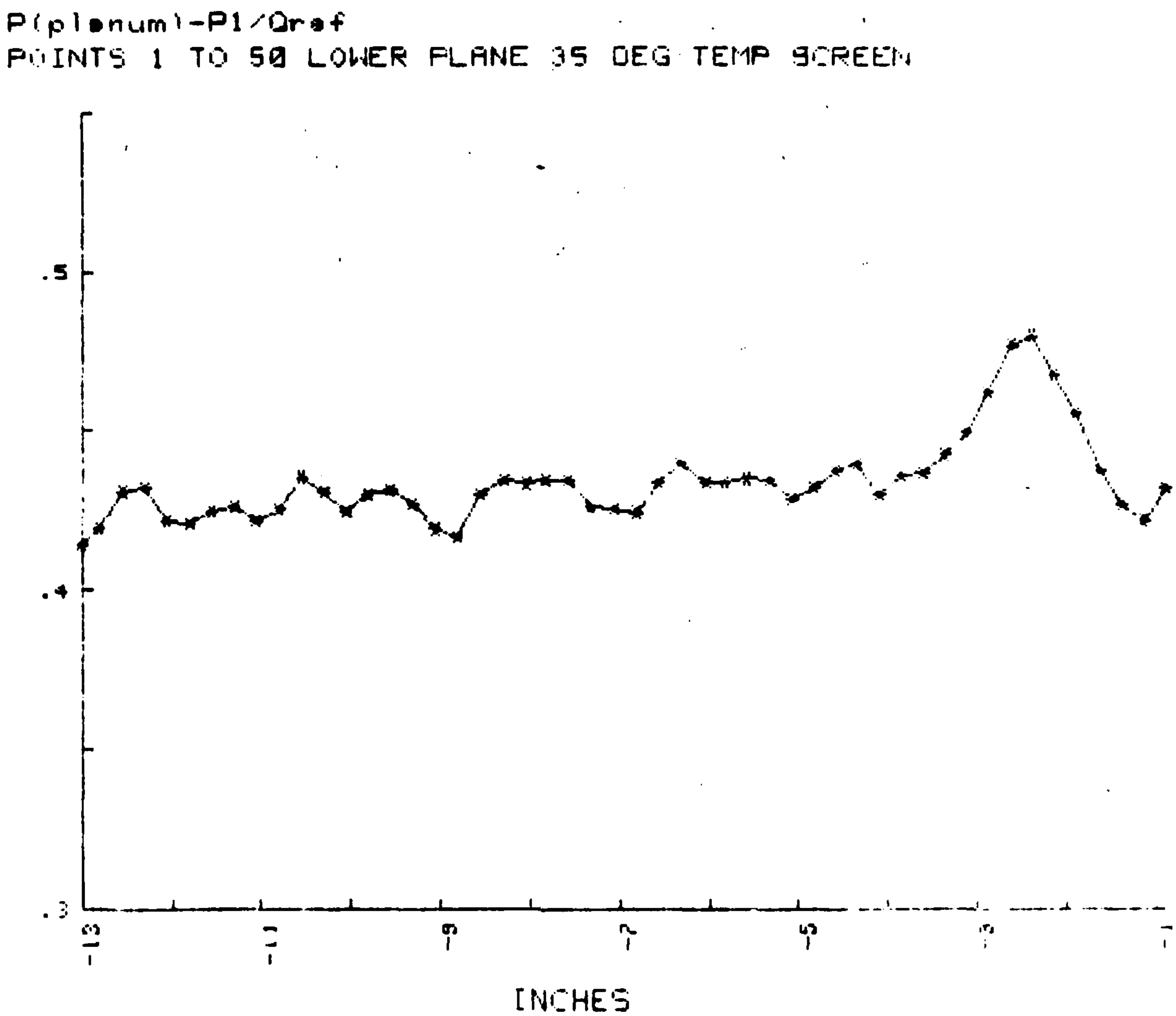


Fig. 34. Probe Survey Data at Midspan, Lower Plane 16 Mesh Screen, Walls at 35°, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

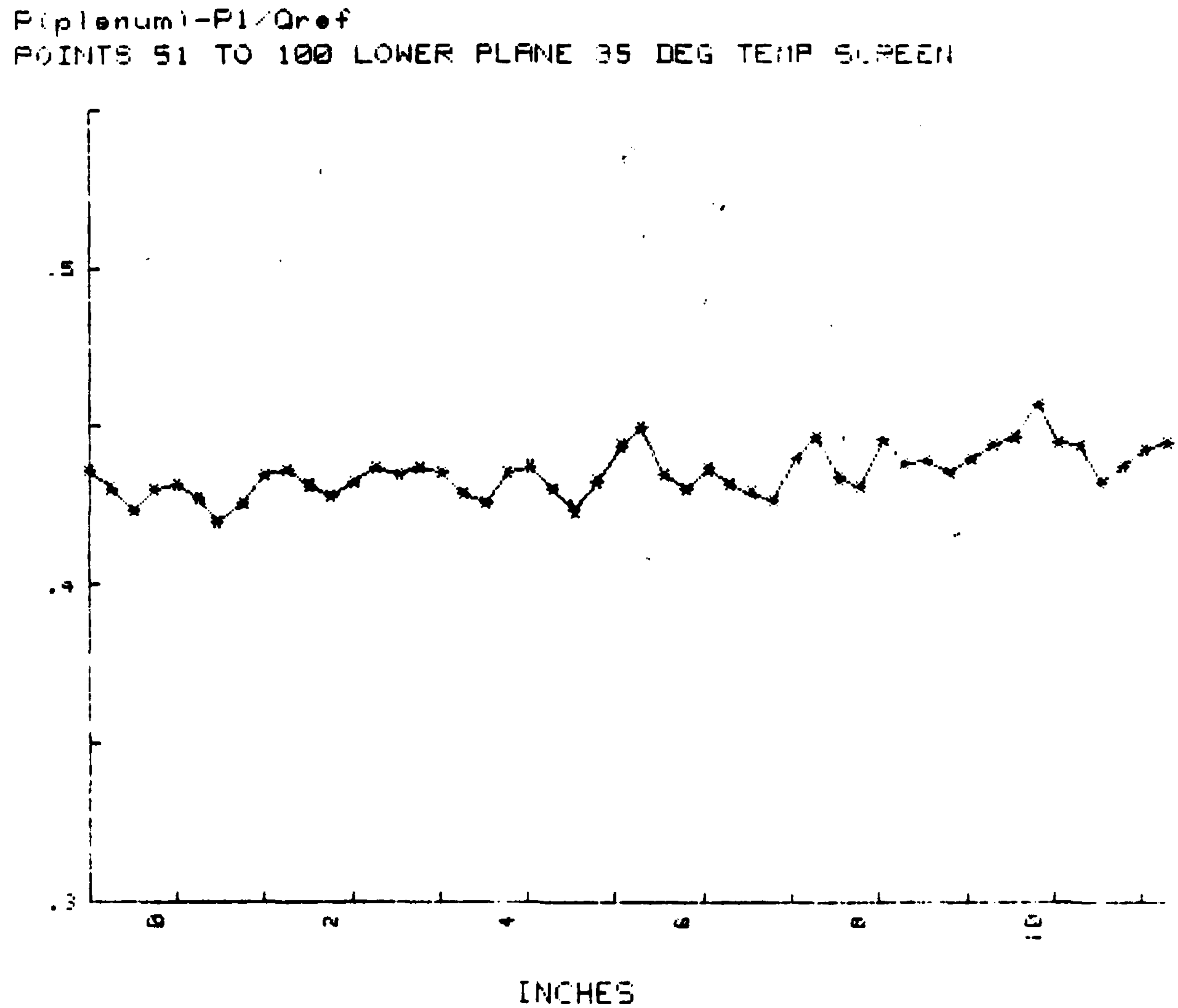


Fig. 35. Probe Survey Data at Midspan, Lower Plane
16 Mesh Screen, Walls at 35°, Points 51 to 100
$(P_{PLENUM} - P_t)/Q_{ref}$

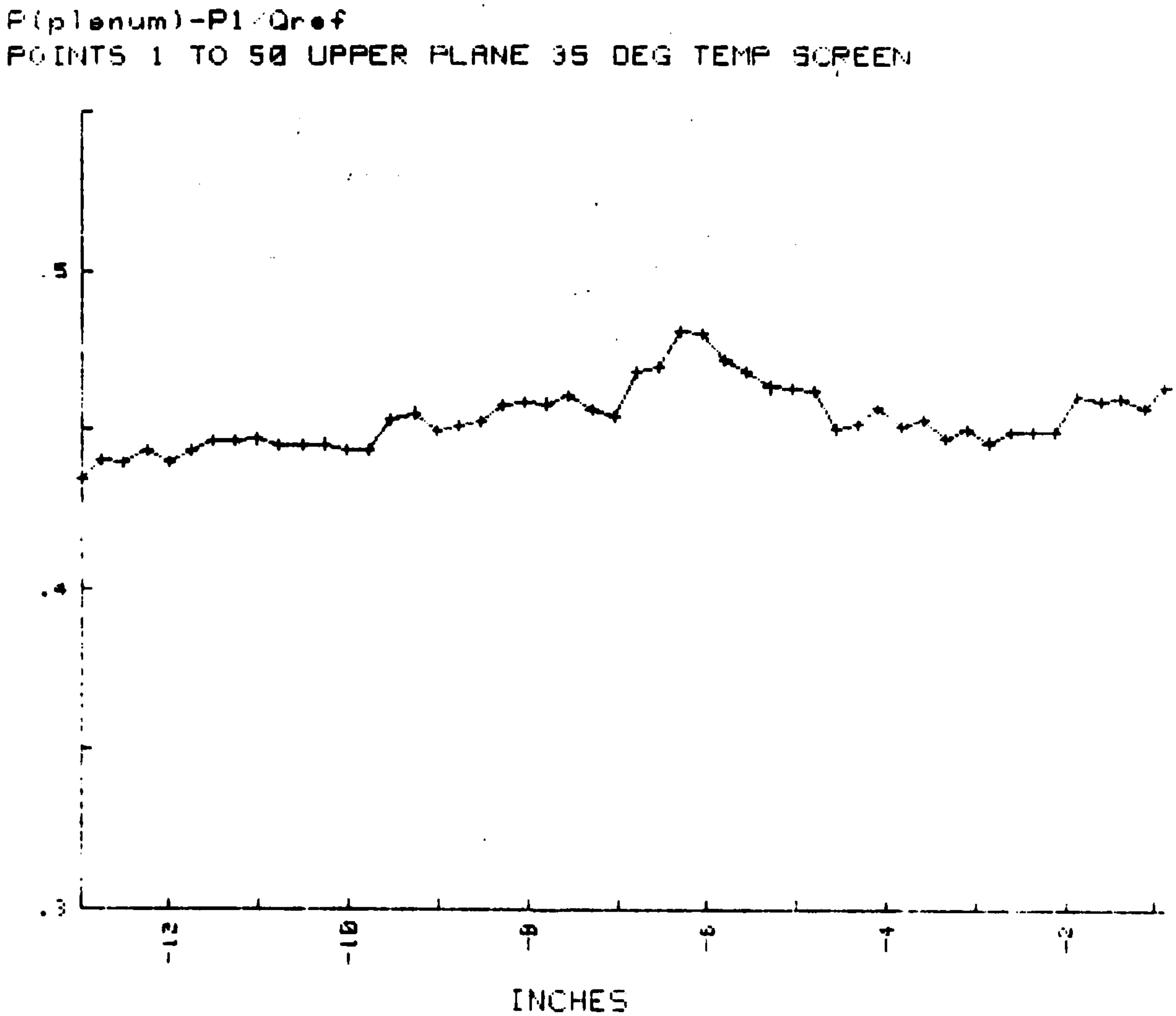


Fig. 36. Probe Survey Data at Midspan, Upper Plane 16 Mesh Screen, Walls at 35°, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

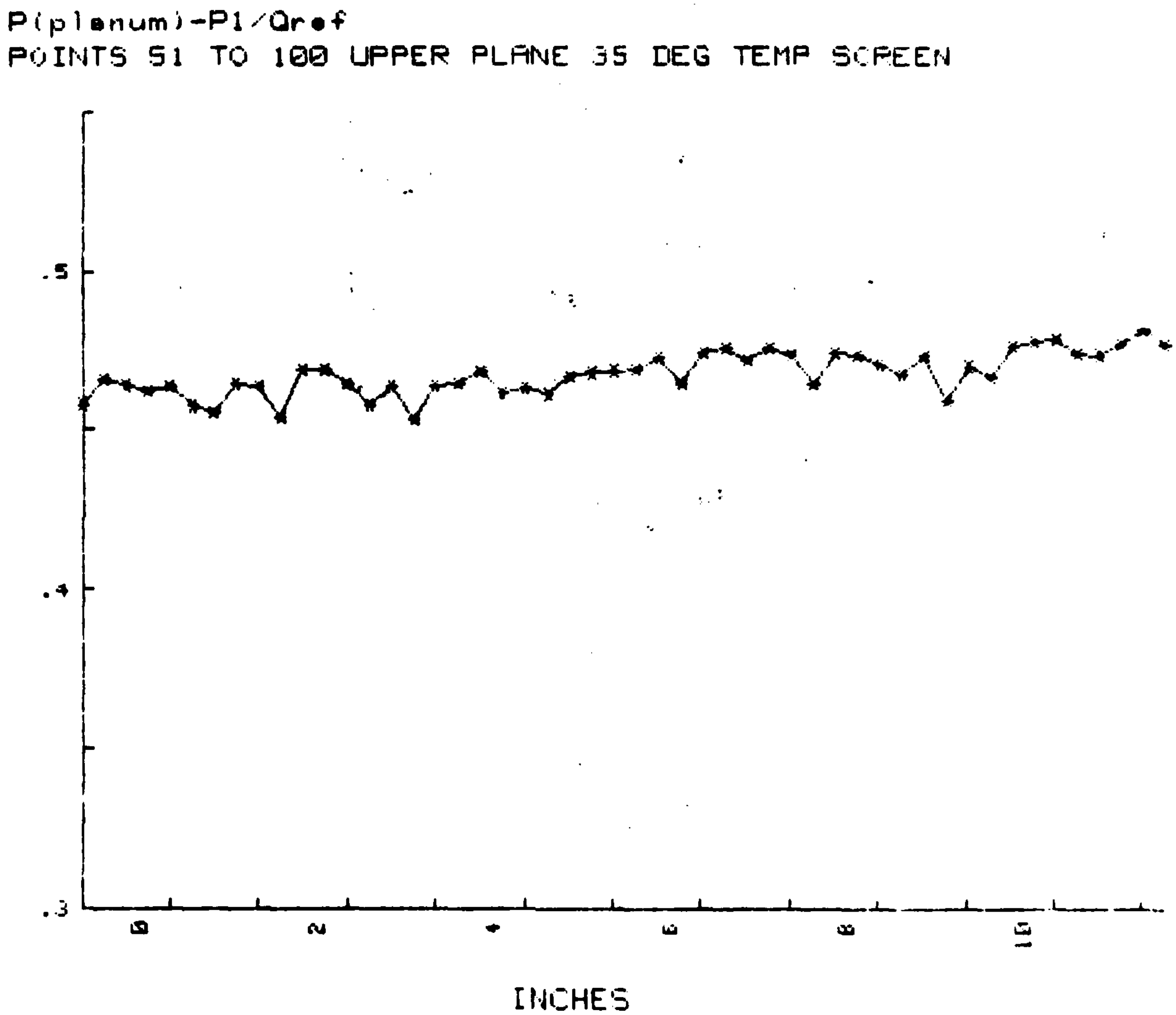


Fig. 37. Probe Survey Data at Midspan, Upper Plane 16 Mesh Screen, Walls at 35°, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

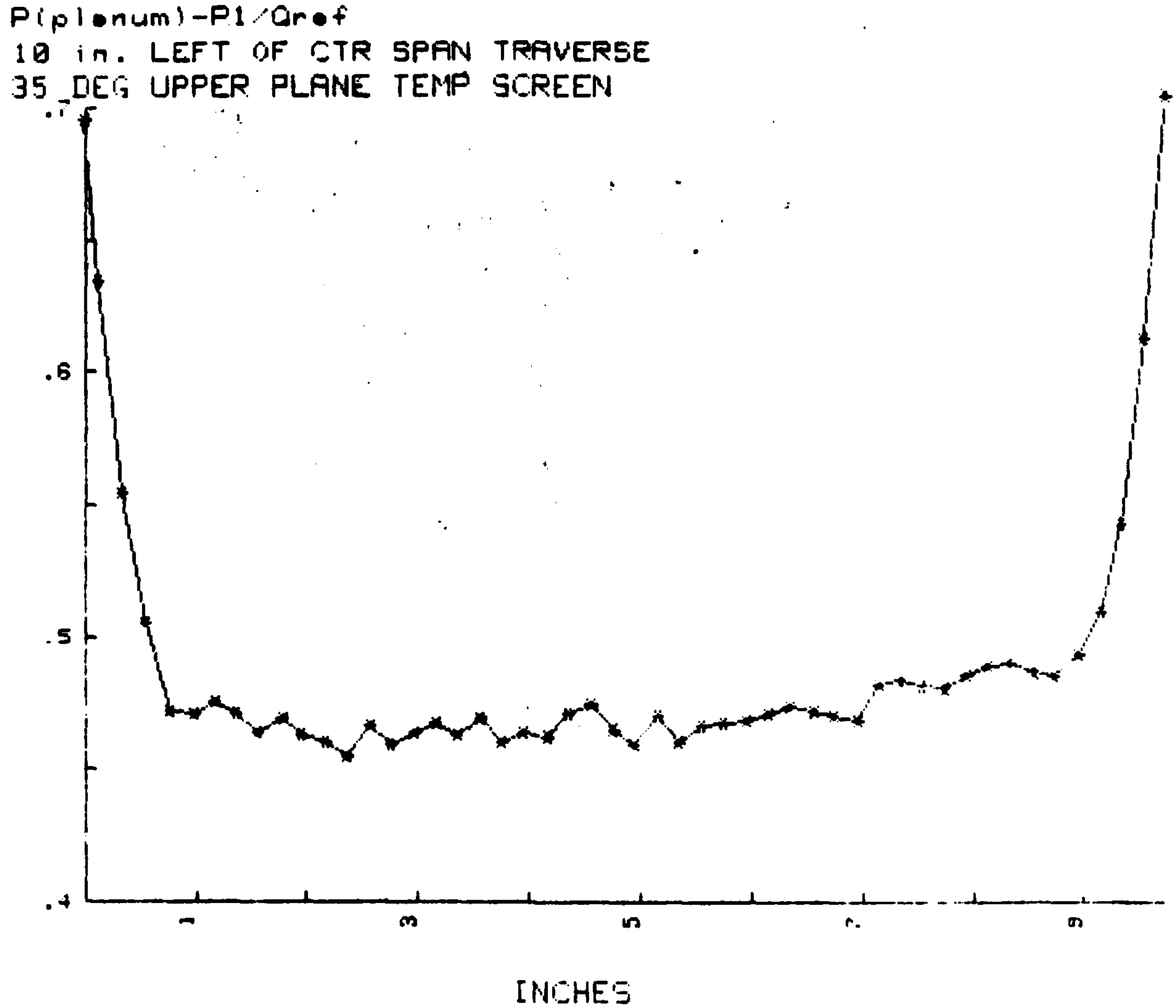


Fig. 38. Probe Survey Data Span Traverse, Lower Plane 16 Mesh Screen, Walls at 35°, 10" Left of CTR $(P_{PLENUM} - P_t)/Q_{ref}$

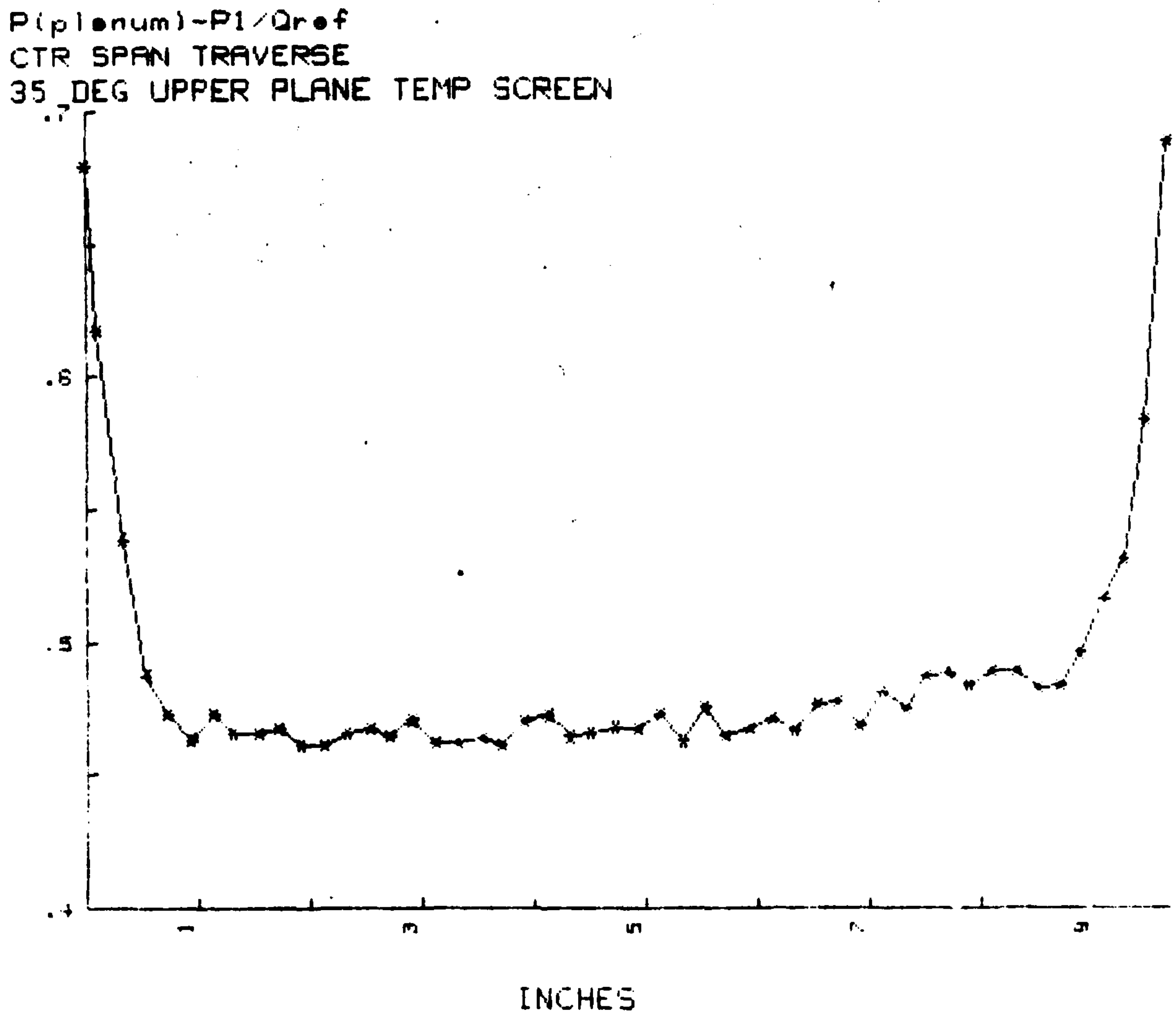


Fig. 39. Probe Survey Data Span Traverse, Lower Plane 16 Mesh Screen, Walls at 35°, Center of Test Section $(P_{PLENUM} - P_t)/Q_{ref}$

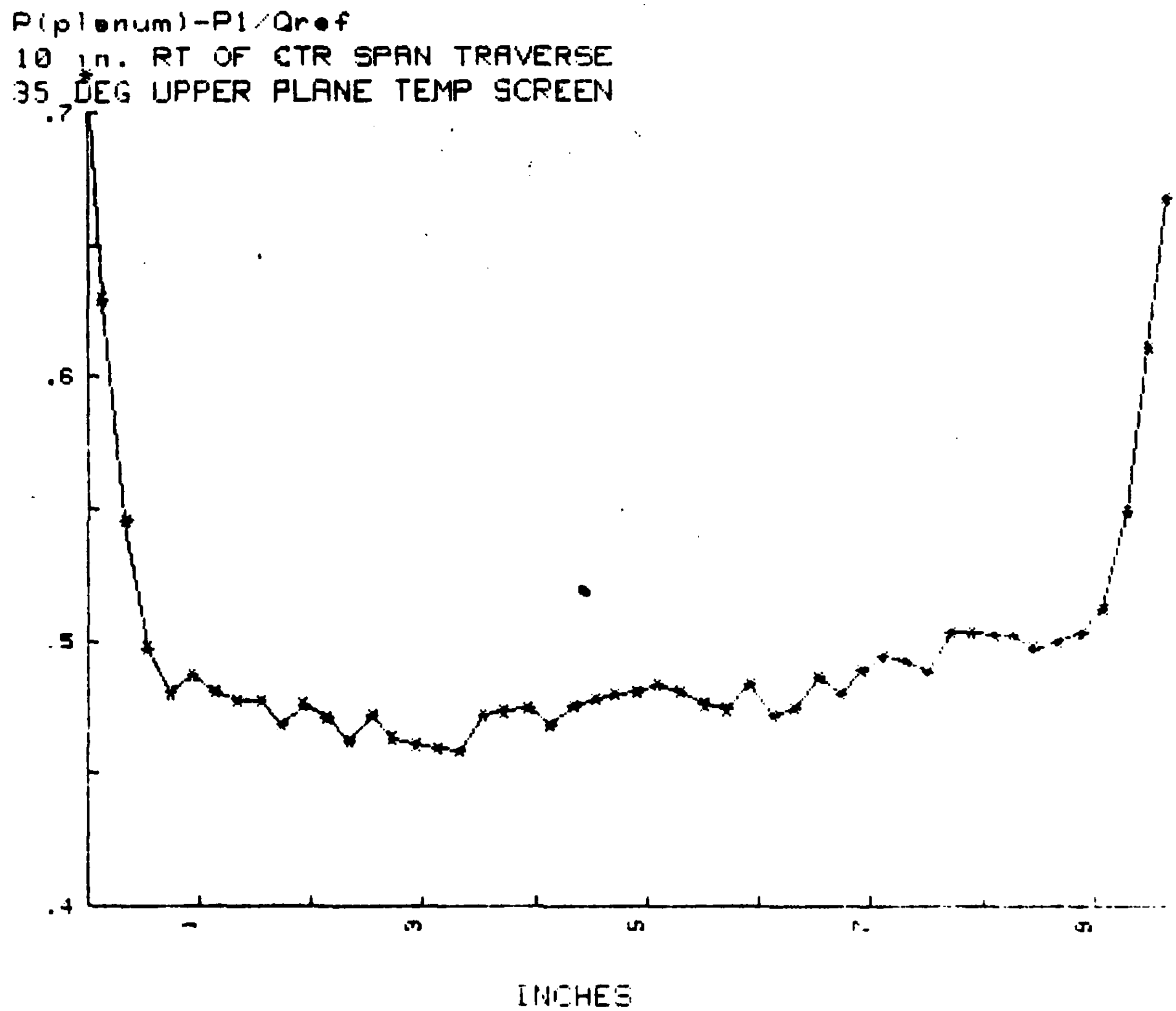


Fig. 40. Probe Survey Data Span Traverse, Lower Plane 16 Mesh Screen, Walls at 35°, Center of Test Section $(P_{PLENUM} - P_t)/Q_{ref}$

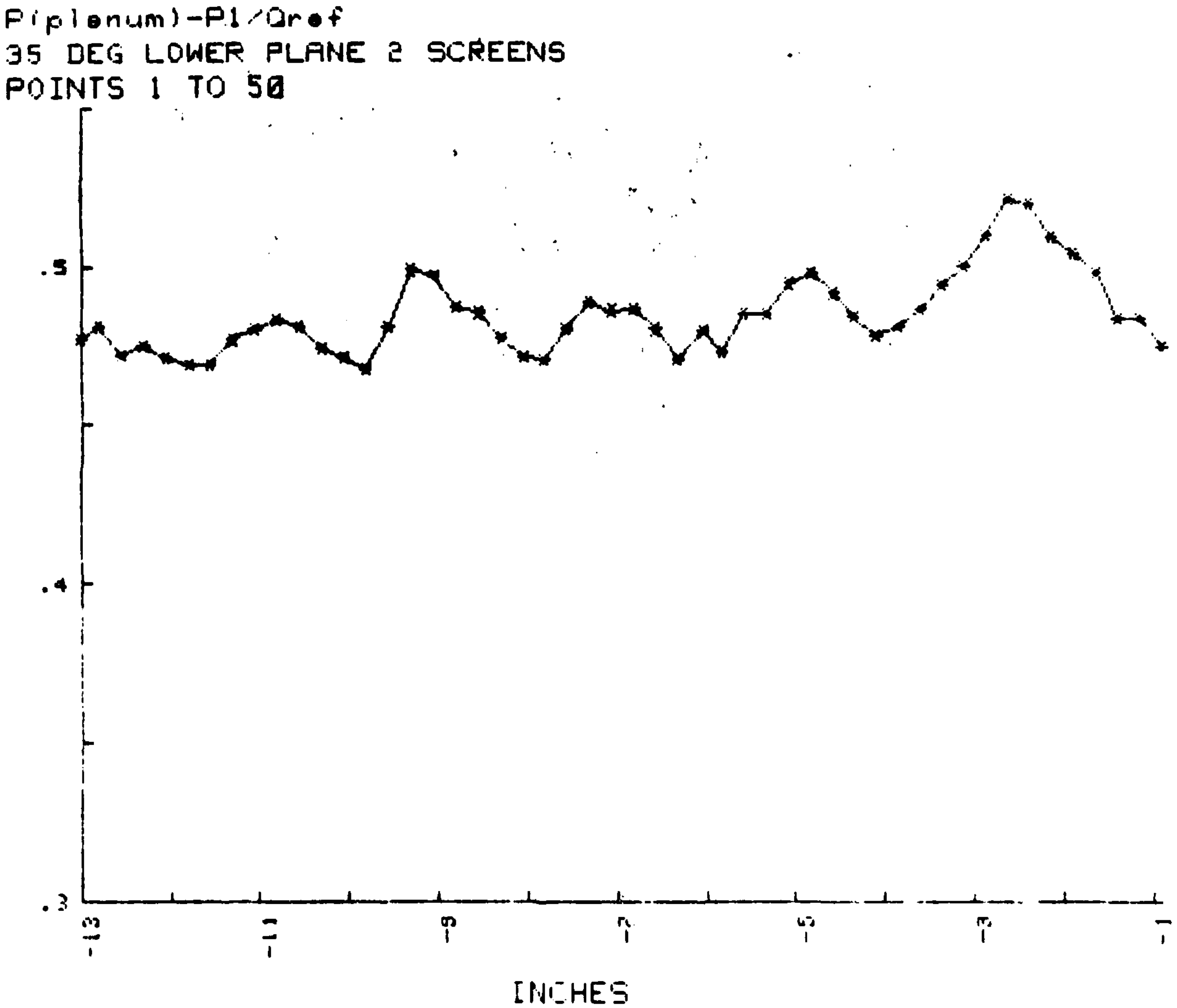
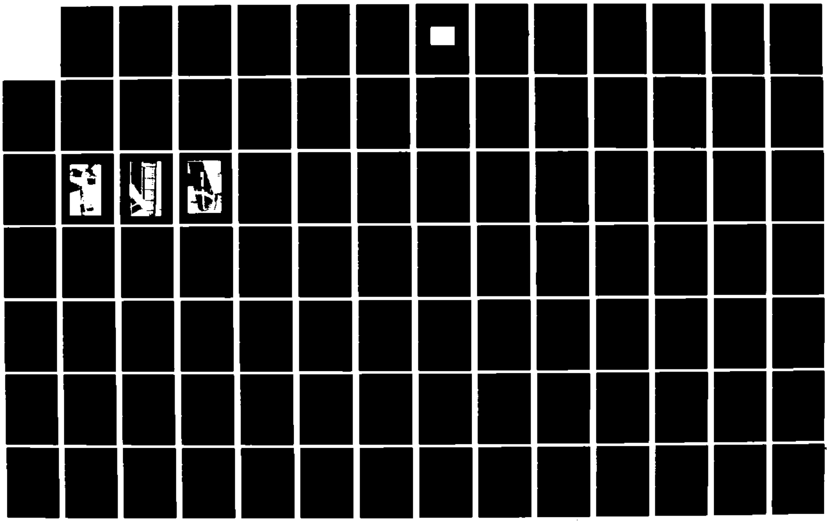


Fig. 41. Probe Survey Data at Midspan, Lower Plane
16 Mesh and 2 Mesh, Walls at 35°, Points 1 to 50
$(P_{PLENUM} - P_t)/Q_{ref}$

OD-A120 456 PRELIMINARY MEASUREMENTS AND CODE CALCULATIONS OF FLOW 2/3
THROUGH A CASCADE OF DCA BLADING AT A SOLIDITY OF 167
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W D MOLLOY

UNCLASSIFIED JUN 82 F/G 21/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

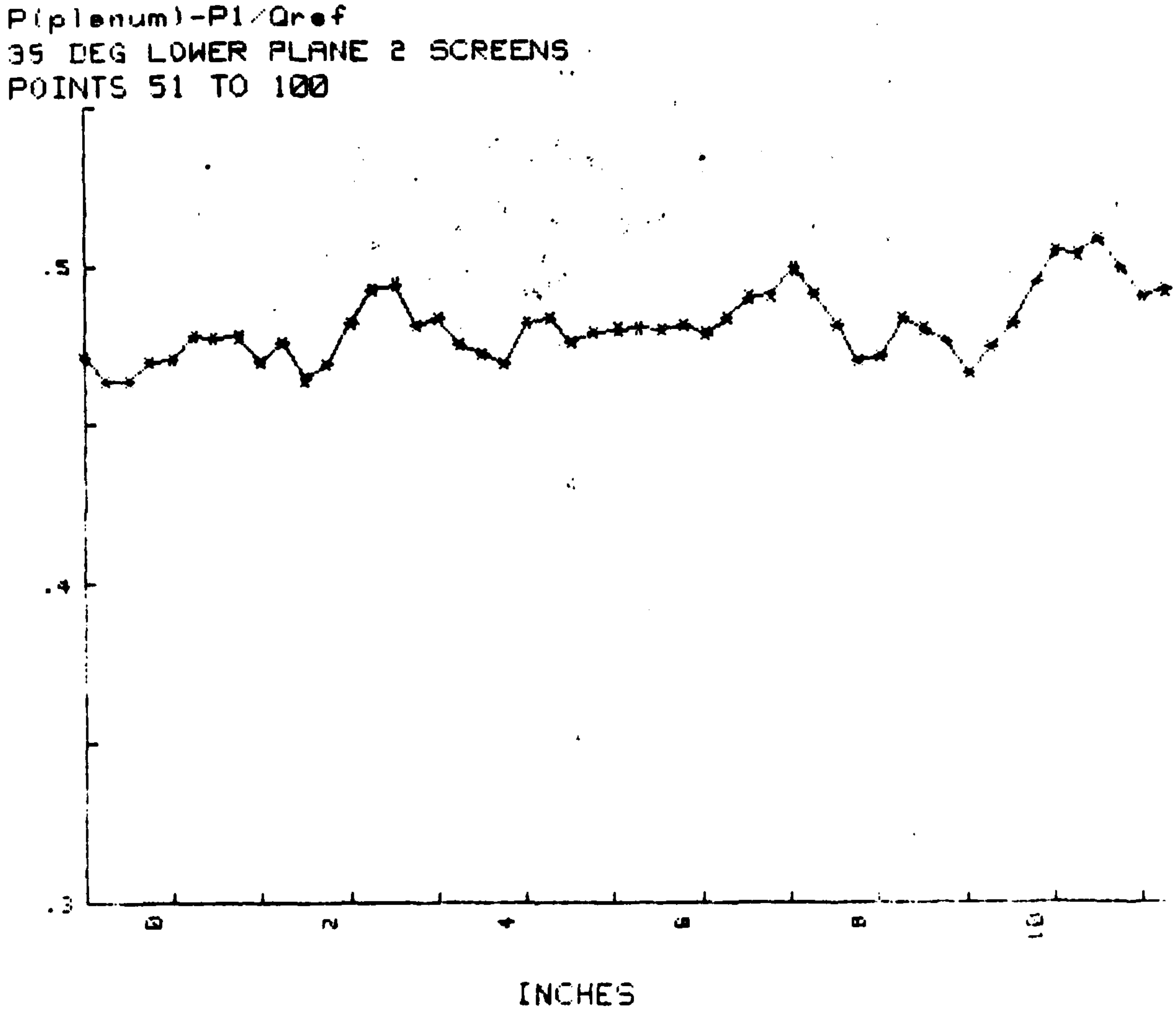


Fig. 42. Probe Survey Data at Midspan, Lower Plane 16 Mesh and 2 Mesh, Walls at 35°, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

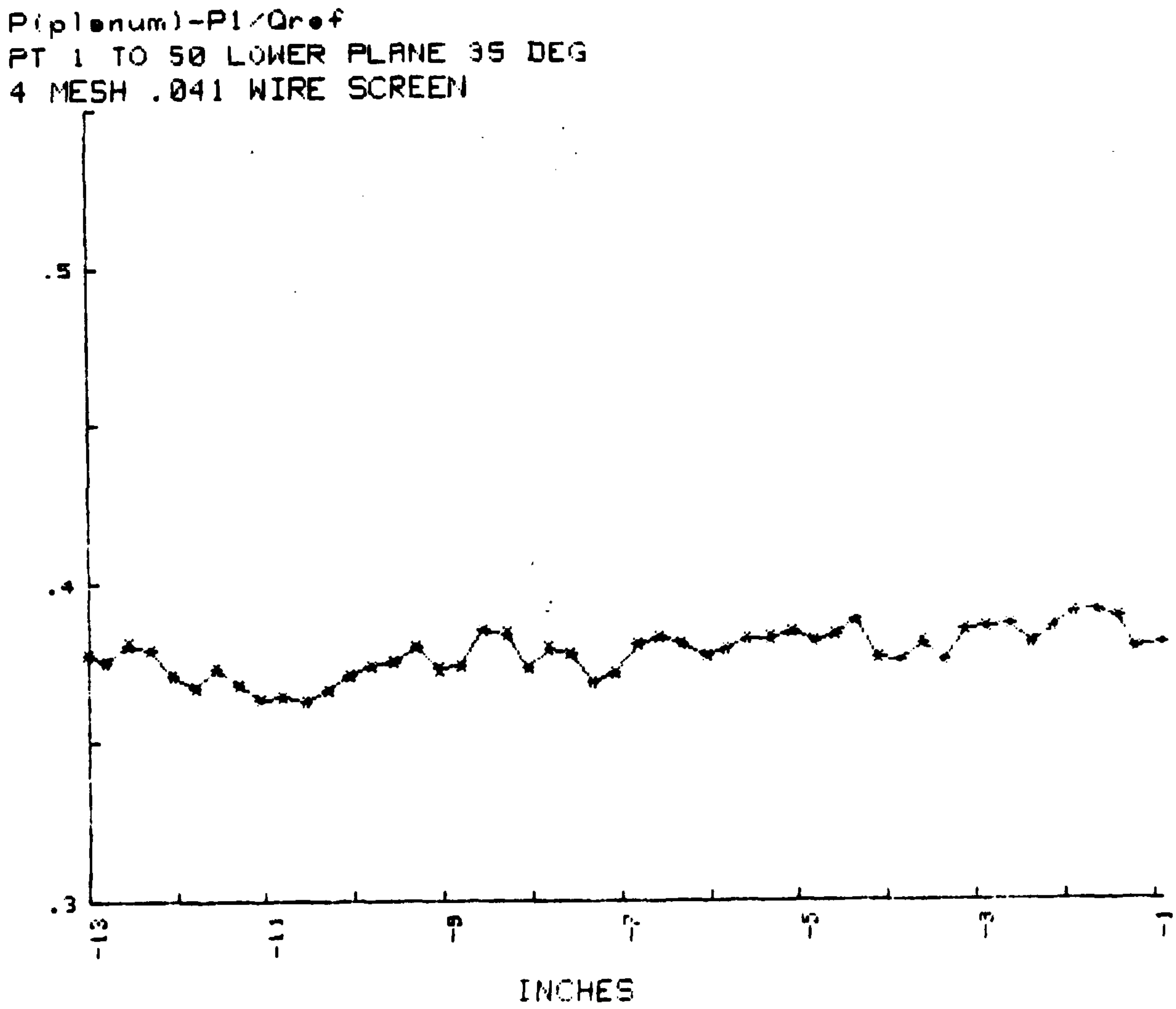


Fig. 43. Probe Survey Data at Midspan, Lower Plane
4 Mesh Screen, Walls at 35°, Points 1 to 50
$(P_{PLENUM} - P_t)/Q_{ref}$

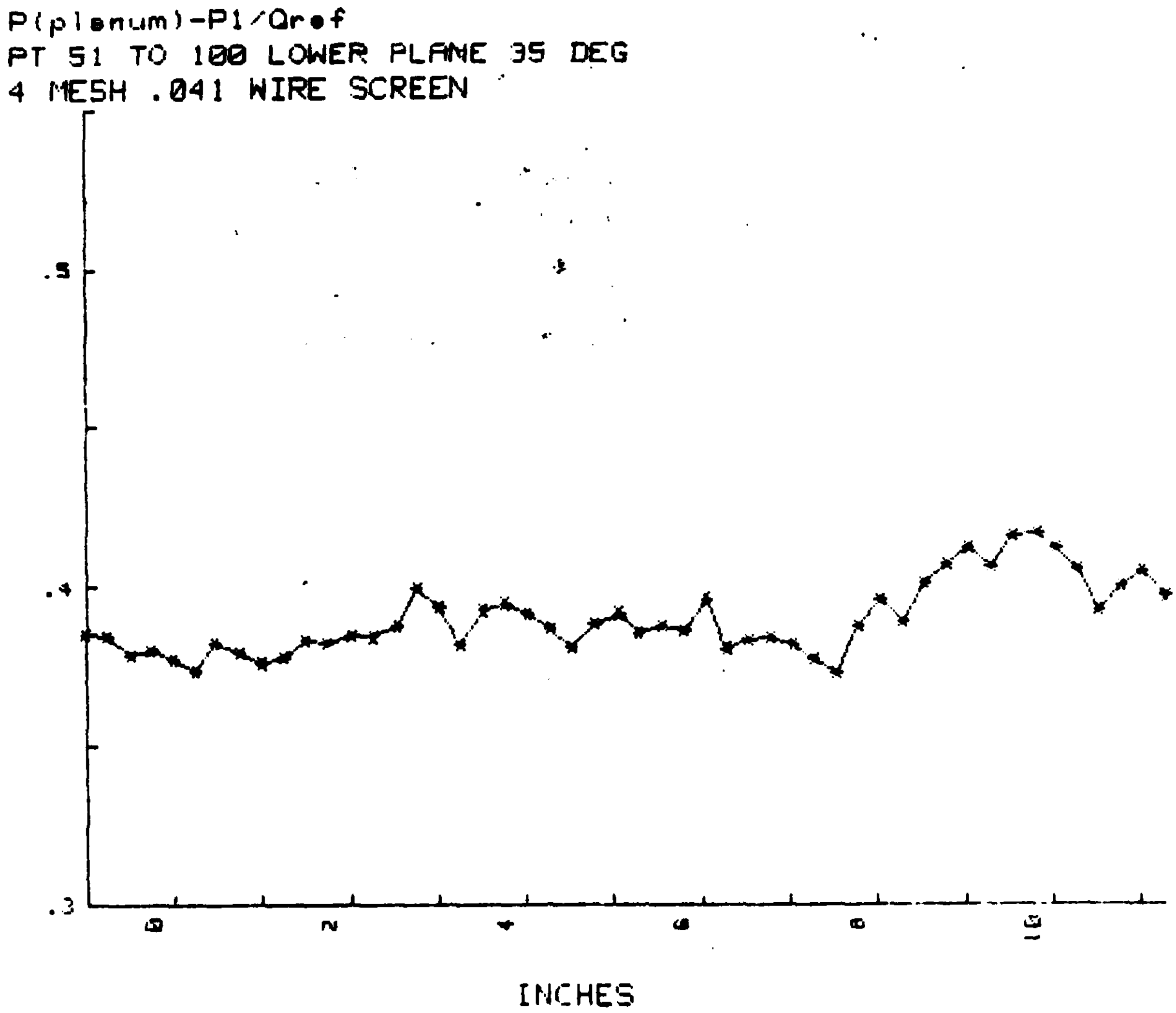


Fig. 44. Probe Survey Data at Midspan, Lower Plane
4 Mesh Screen, Walls at 35°, Points 51 to 100
$(P_{PLENUM} - P_t)/Q_{ref}$

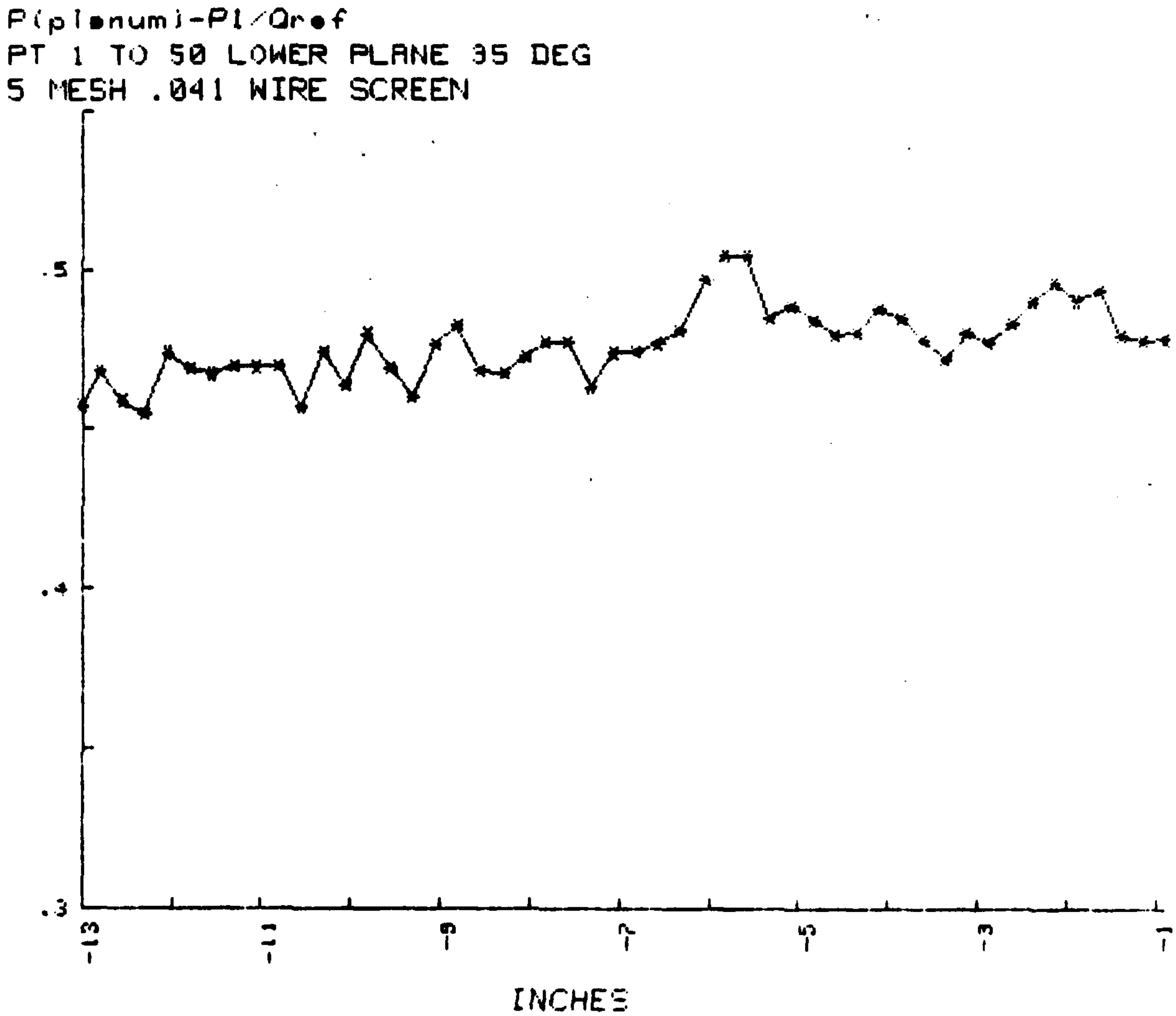


Fig. 45. Probe Survey Data at Midspan, Lower Plane 5 Mesh Screen, Walls at 35°, Points 1 to 50 $(P_{PLENUM} - P_t)/Q_{ref}$

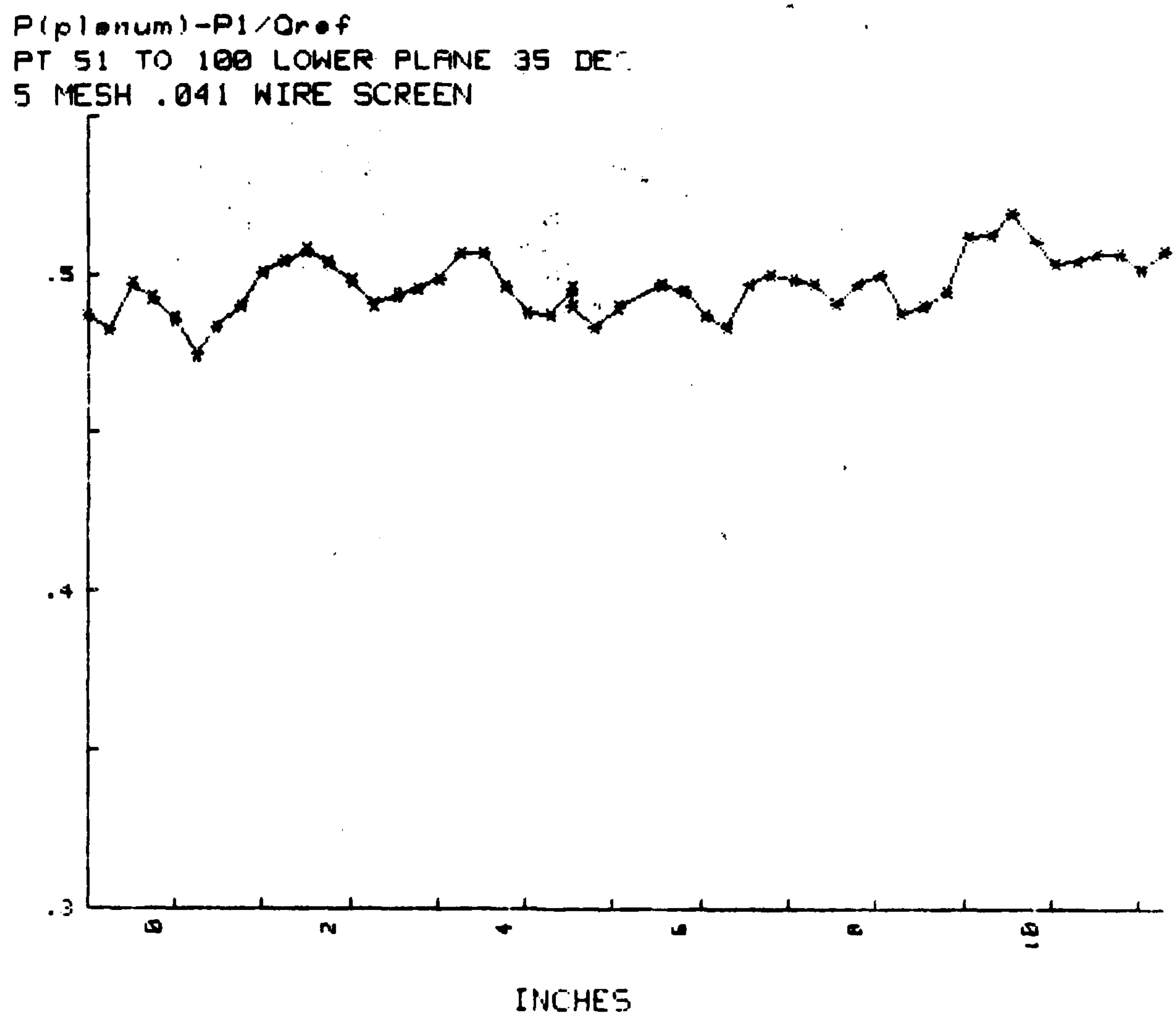


Fig. 46. Probe Survey Data at Midspan, Lower Plane 5 Mesh Screen, Walls at 35°, Points 51 to 100 $(P_{PLENUM} - P_t)/Q_{ref}$

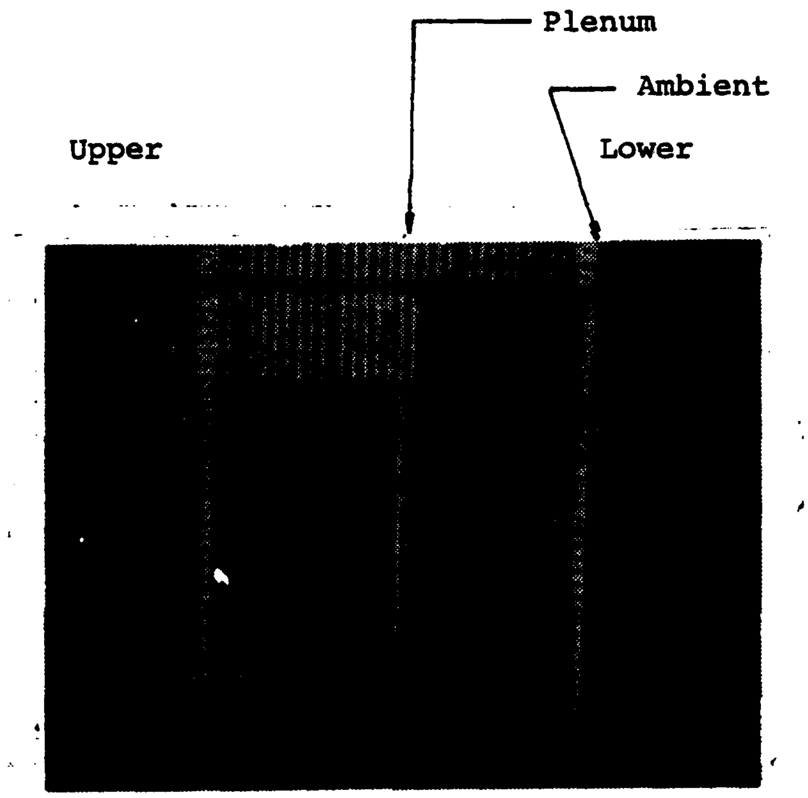


Fig. 47. Wall Static Pressure Distribution

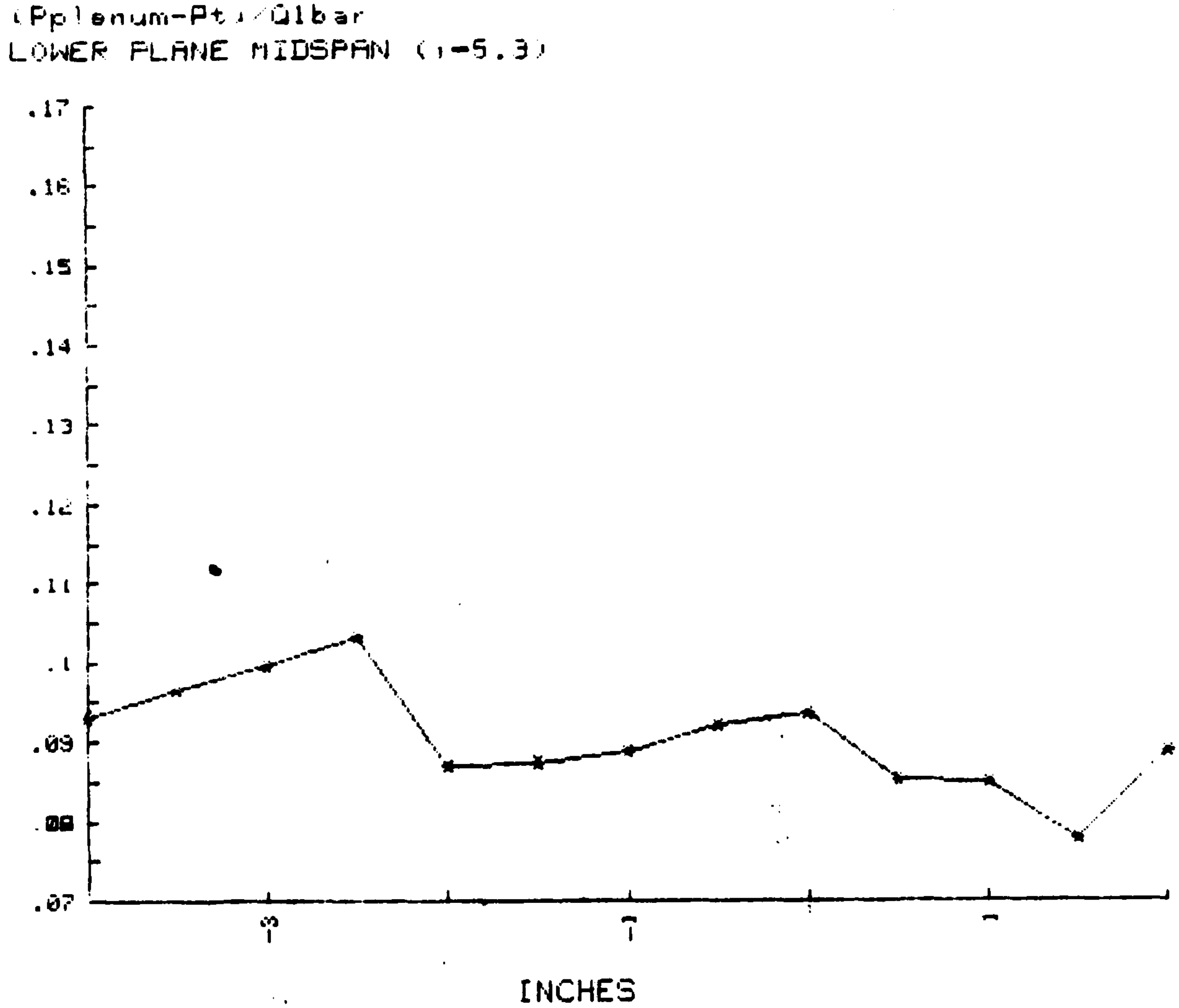
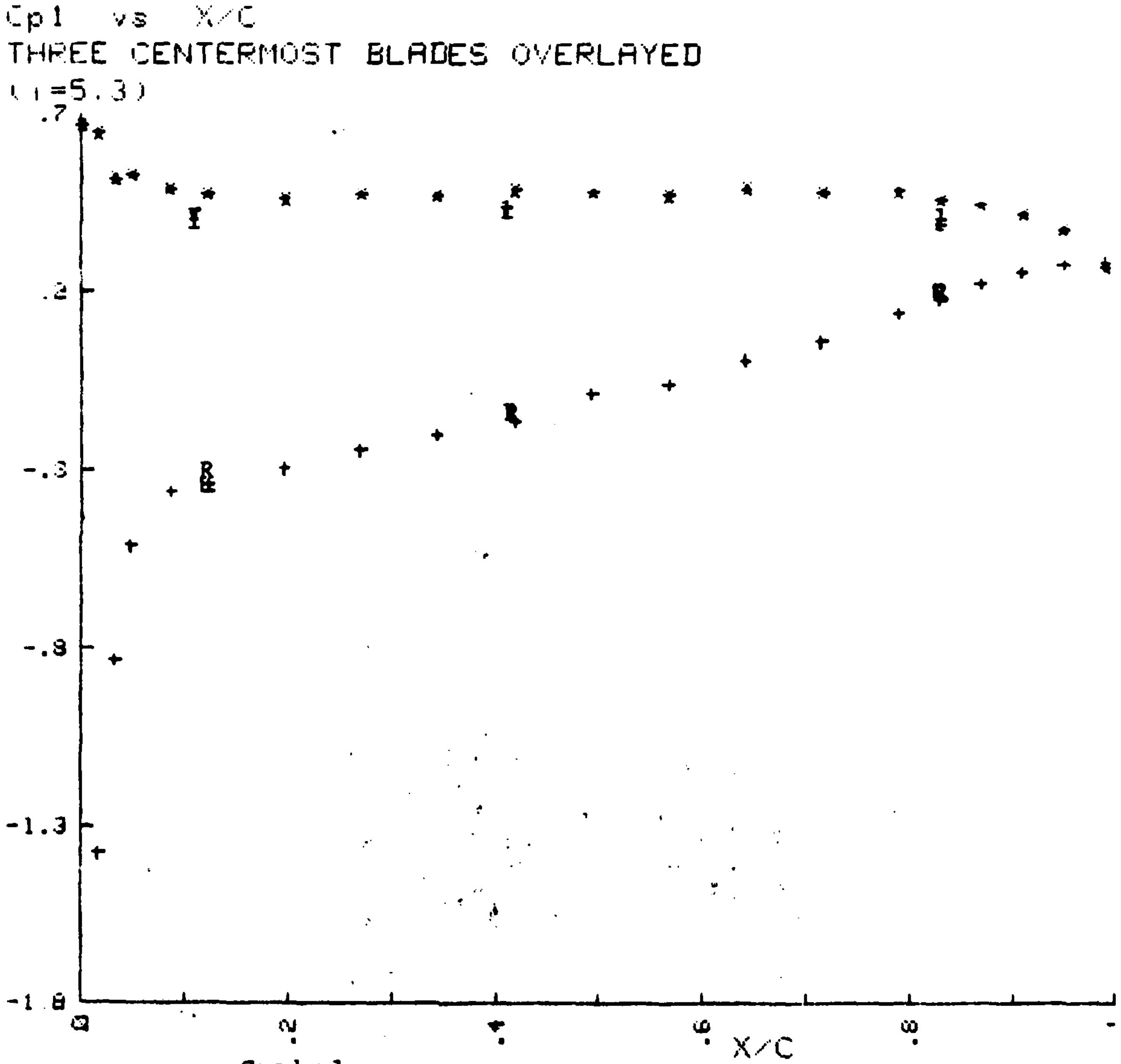


Fig. 48. Probe Survey Data at Upstream Midspan ($i = 5.3$, $(P_{PLENUM} - P_t)/\bar{Q}_1$, Lower Plane)



Symbols:

BLADE	LEFT	CENTER	RIGHT
Pressure Side	l	*	r
Suction Side	L	+	R

Fig. 49. Blade Surface Pressure Distribution on Three Centermost Blades (i = 5.3)

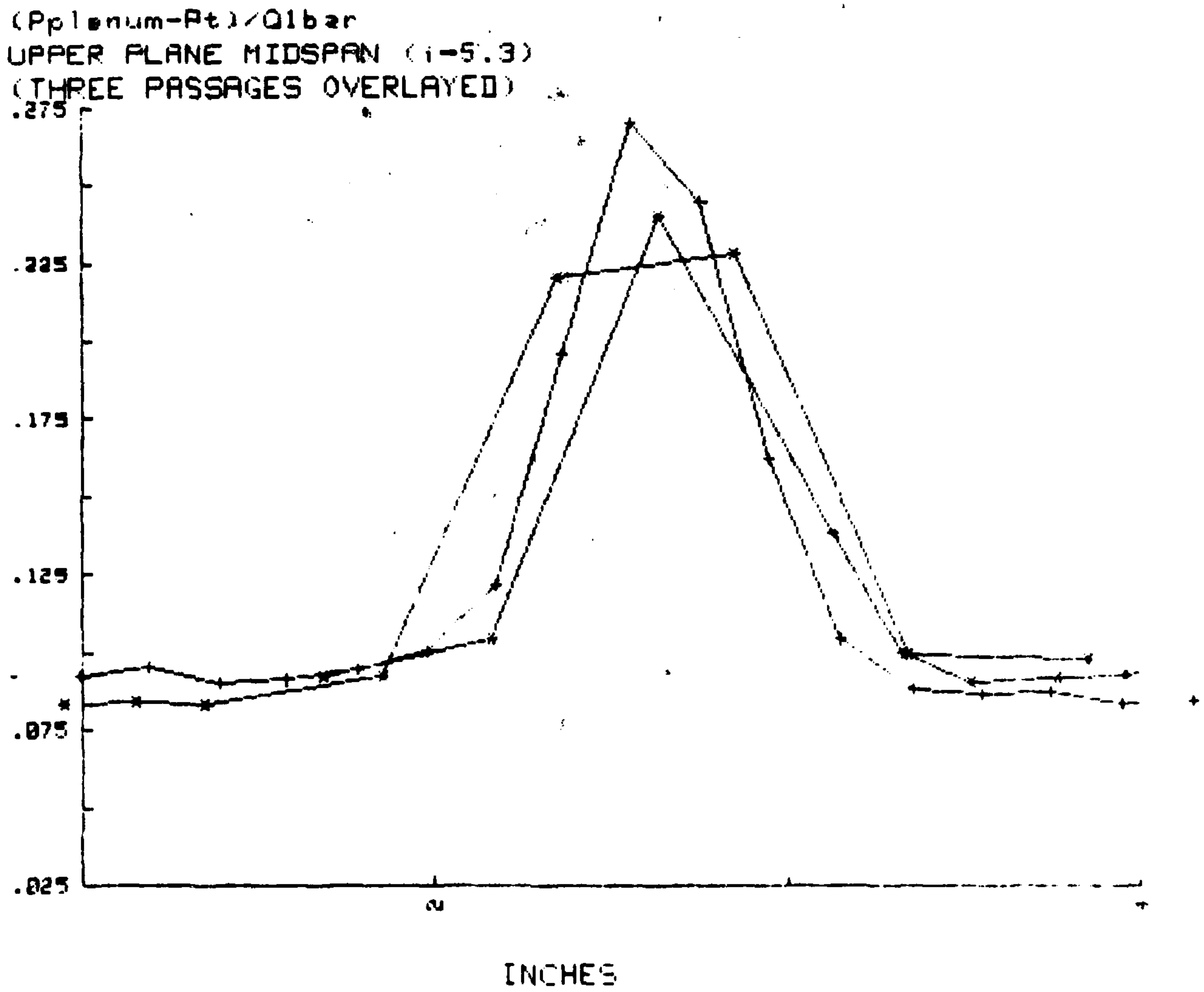


Fig. 50. Probe Survey Data at Midspan
(i = 5.3, $(P_{PLENUM} - P_t)/\bar{Q}_1$, Upper Plane)

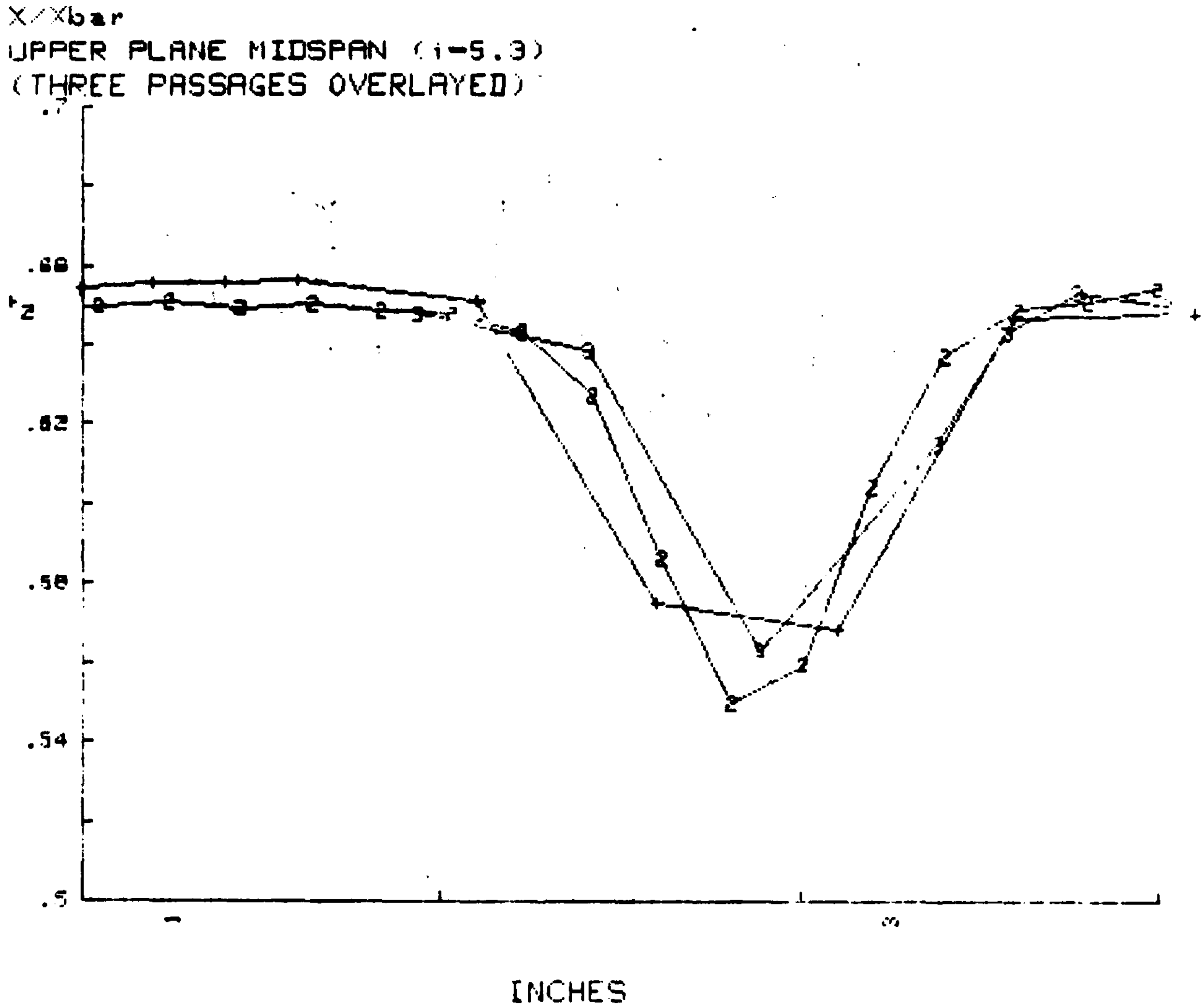


Fig. 51. Probe Survey Data at Midspan
(i = 5.3, ($X/\overline{X}$), Upper Plane)

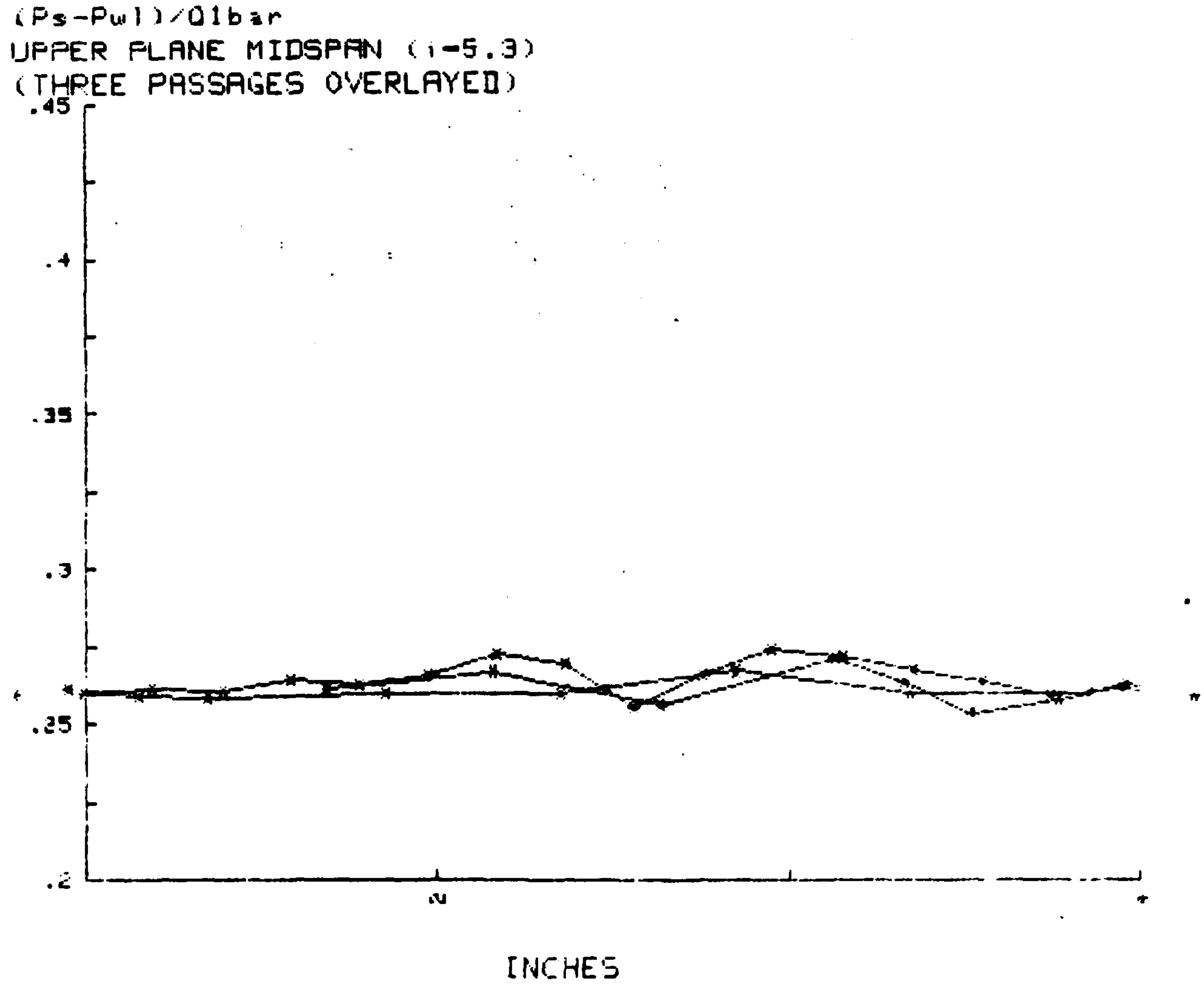


Fig. 52. Probe Survey Data at Midspan
(i = 5.3, $(P_s - P_{w1})/\bar{Q}_1$, Upper Plane)

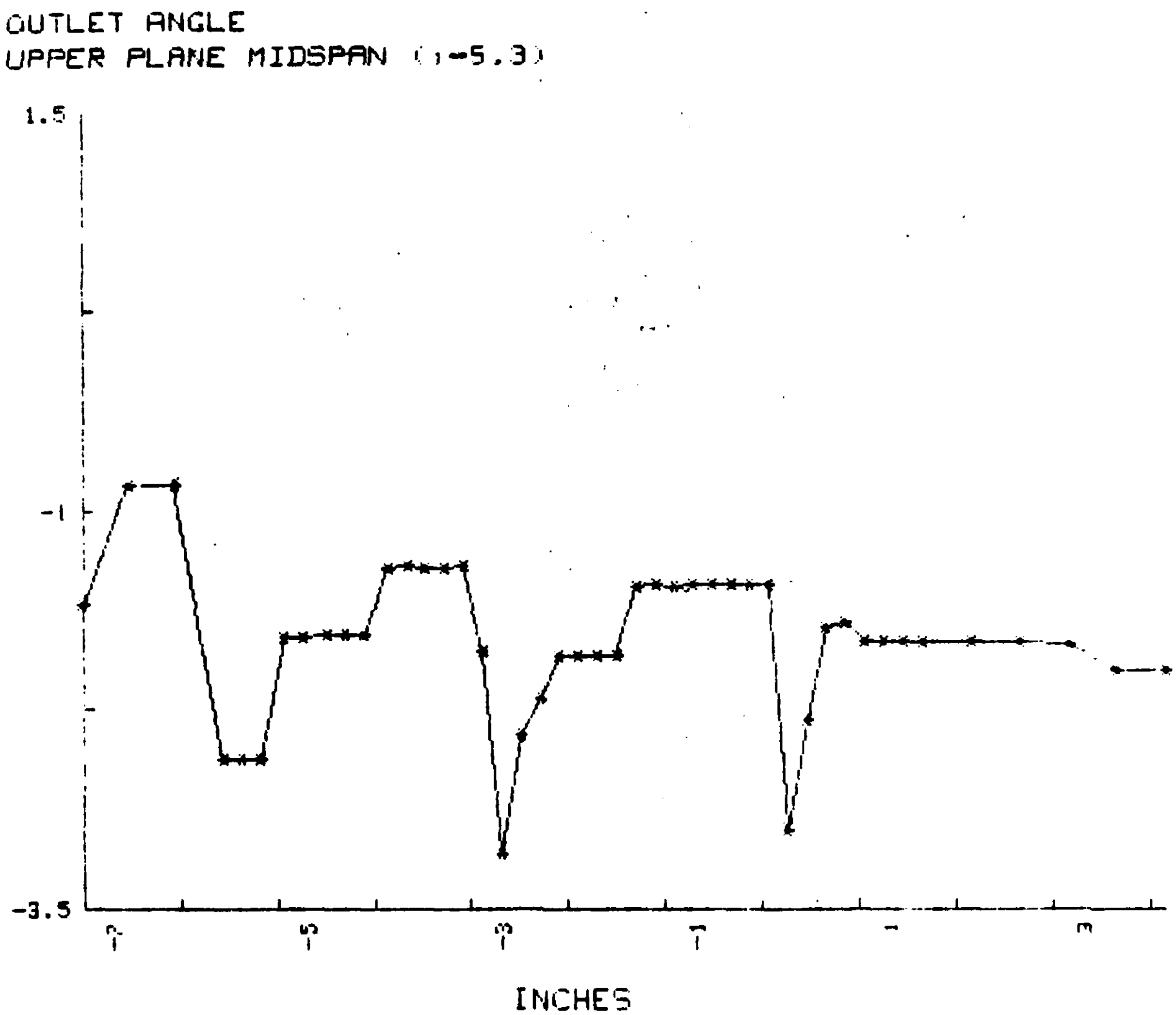


Fig. 53. Probe Survey Data at Midspan
(i = 5.3, Outlet Angle, Upper Plane)

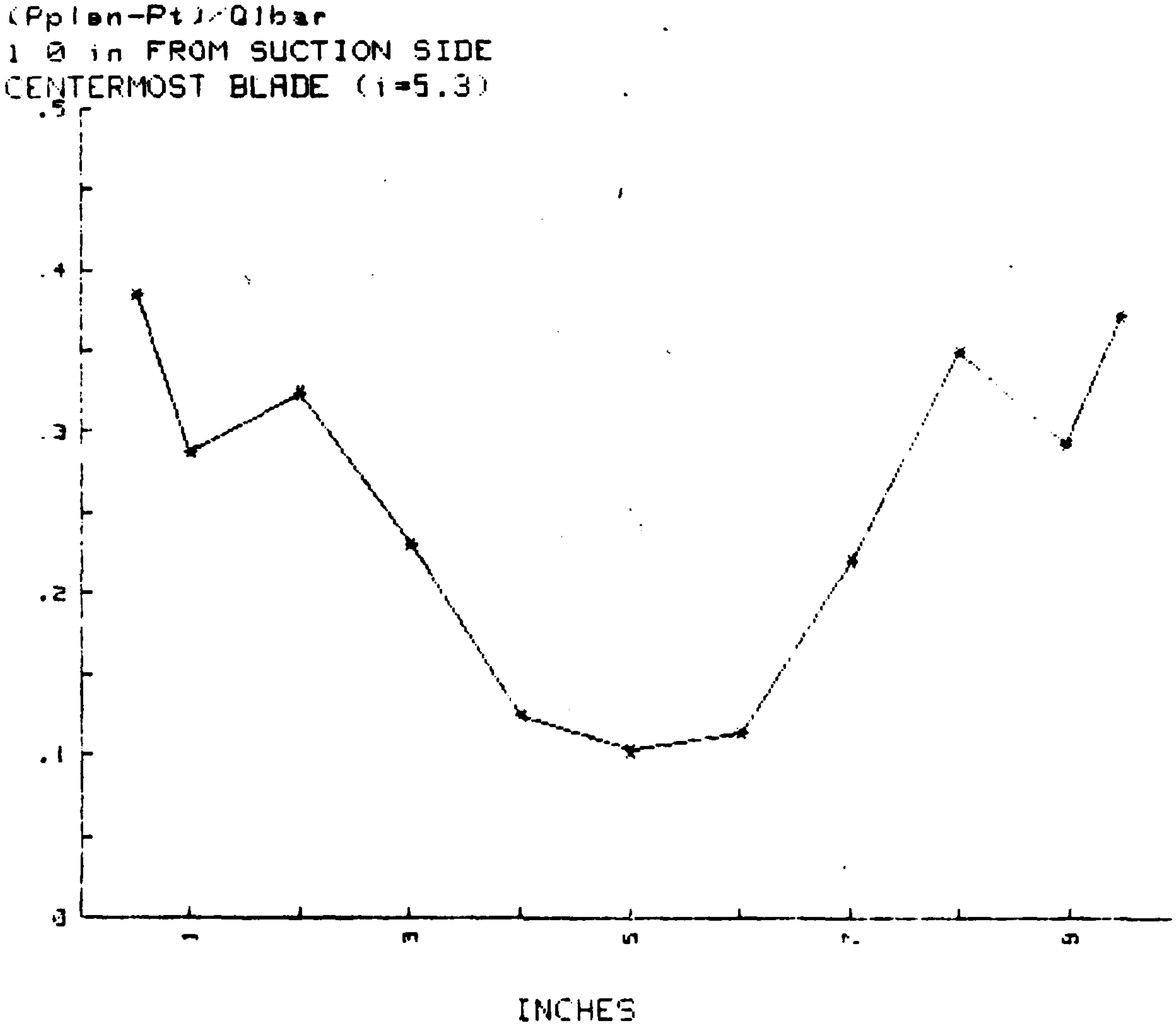


Fig. 54. Spanwise Probe Data Surveyed 1 in. from Suction Side of Centermost Blade ($i = 5.3$, $(P_{PLENUM} - P_t)/\bar{Q}_1$, Upper Plane)

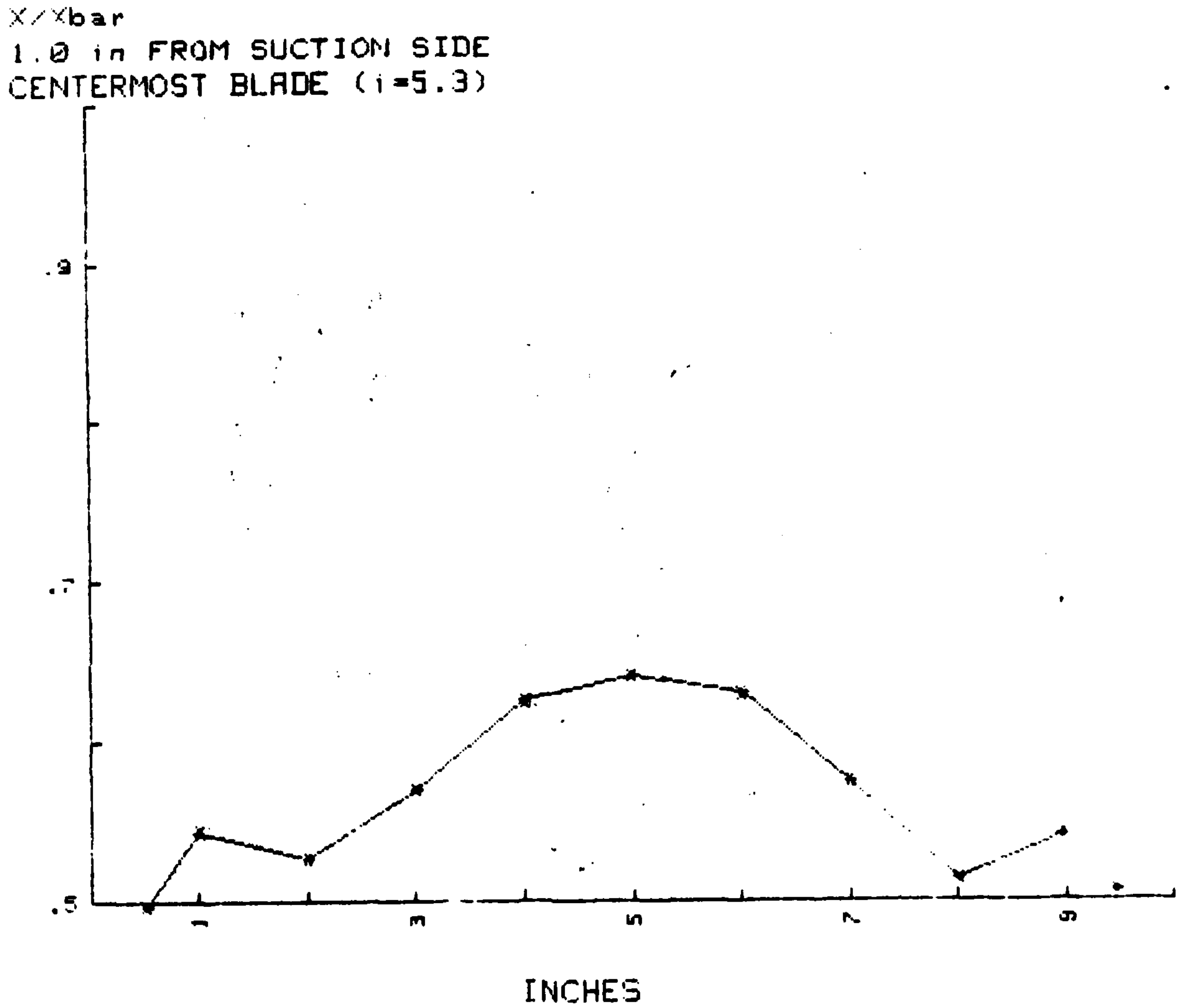


Fig. 55. Spanwise Probe Data Surveyed 1 in. from Suction Side of Centermost Blade (i = 5.3, $X/\bar{X}$, Upper Plane)

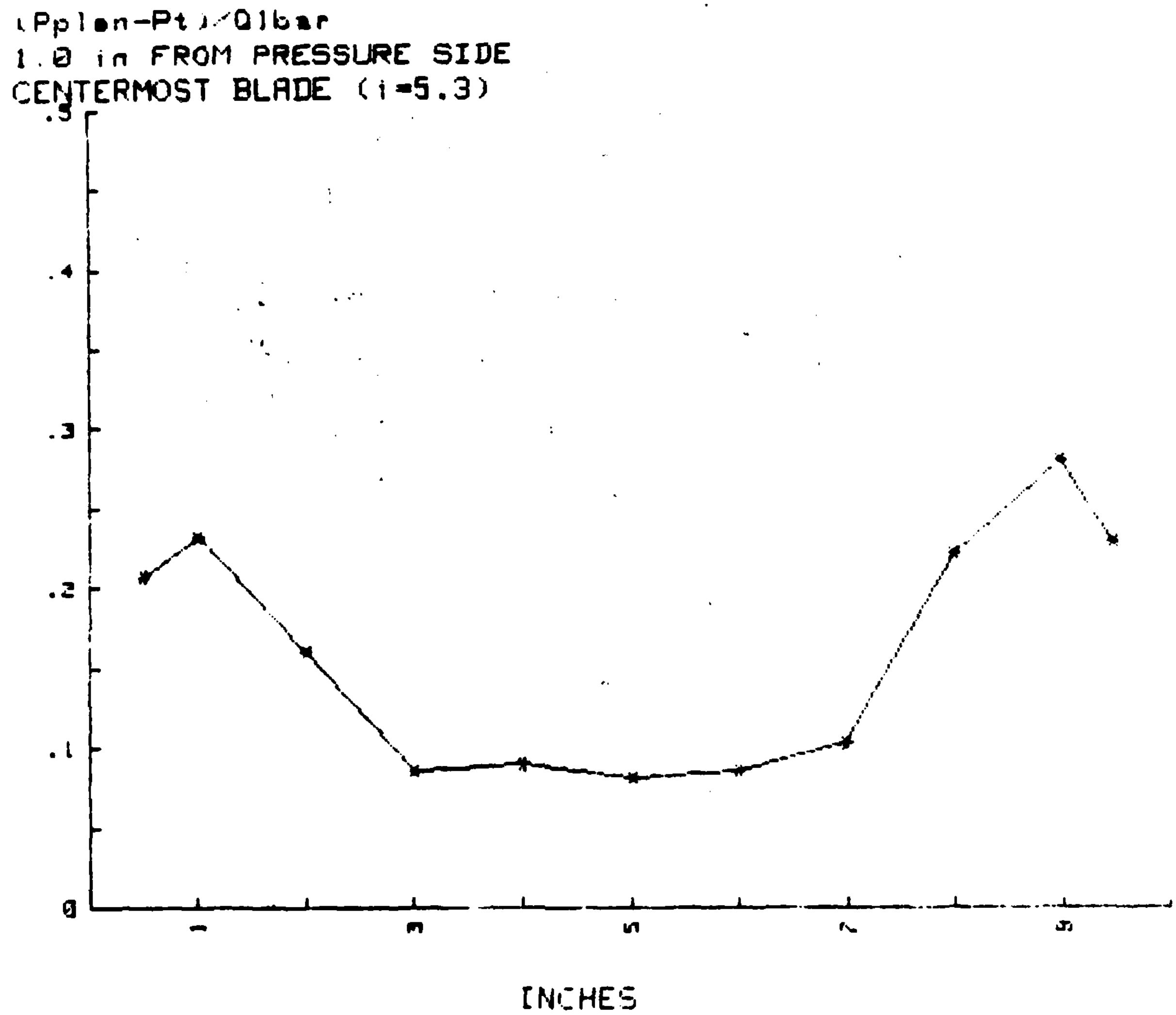


Fig. 56. Spanwise Probe Data Surveyed 1 in. from Pressure Side of Centermost Blade ($i = 5.3$, $(P_{PLENUM} - P_t)/\overline{Q}_1$, Upper Plane)

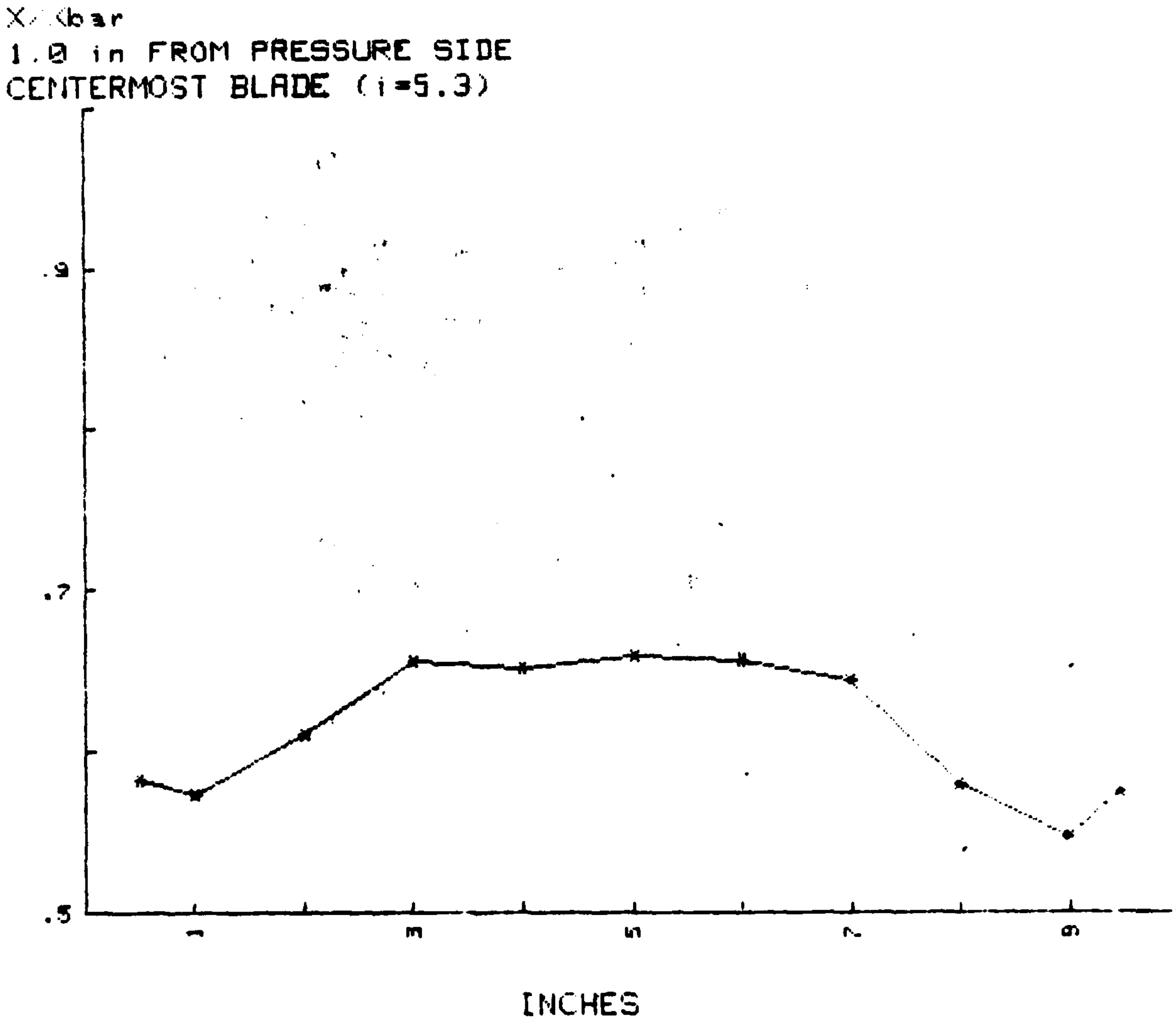


Fig. 57. Spanwise Probe Data Surveyed 1 in. from Pressure Side of Centermost Blade (i = 5.3, $X/\overline{X}$, Upper Plane)

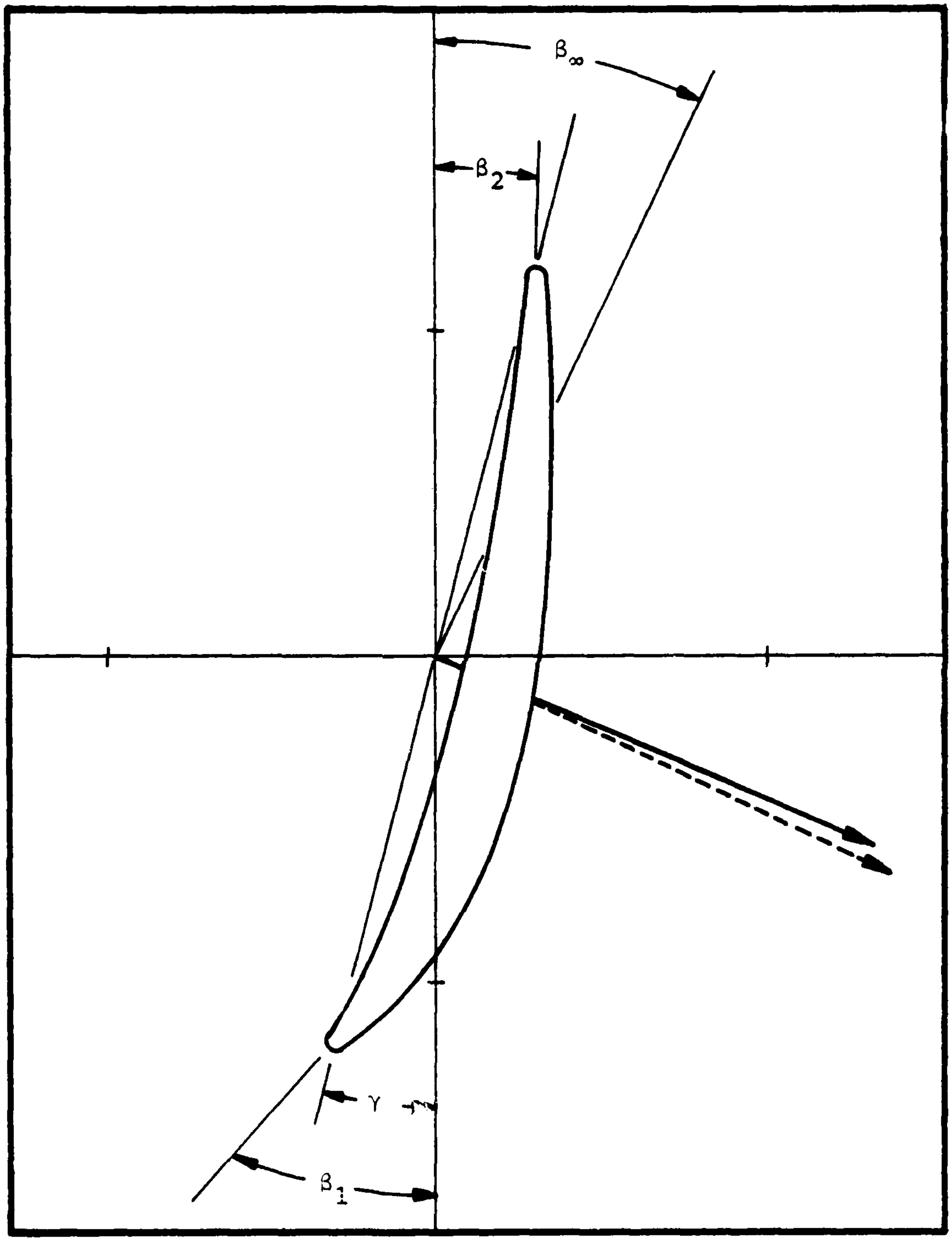


Fig. 58. Resultant Blade Force Vectors by Momentum Balance (– – – – –) and from Surface Pressure Integration (———) i = 5.3

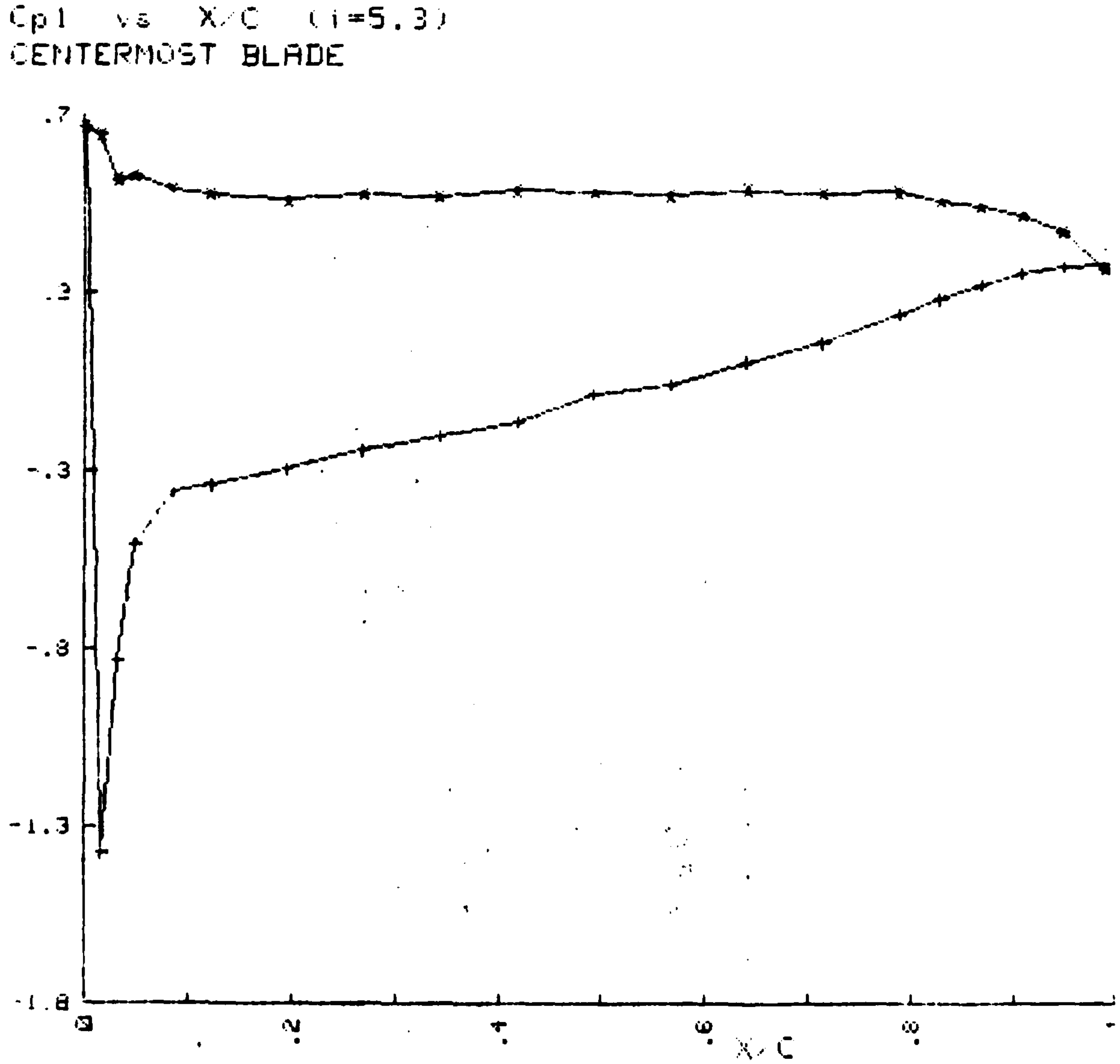


Fig. 59. Measured Blade Surface Pressure Distribution
(i = 5.3, * = Pressure Side,
+ = Suction Side)

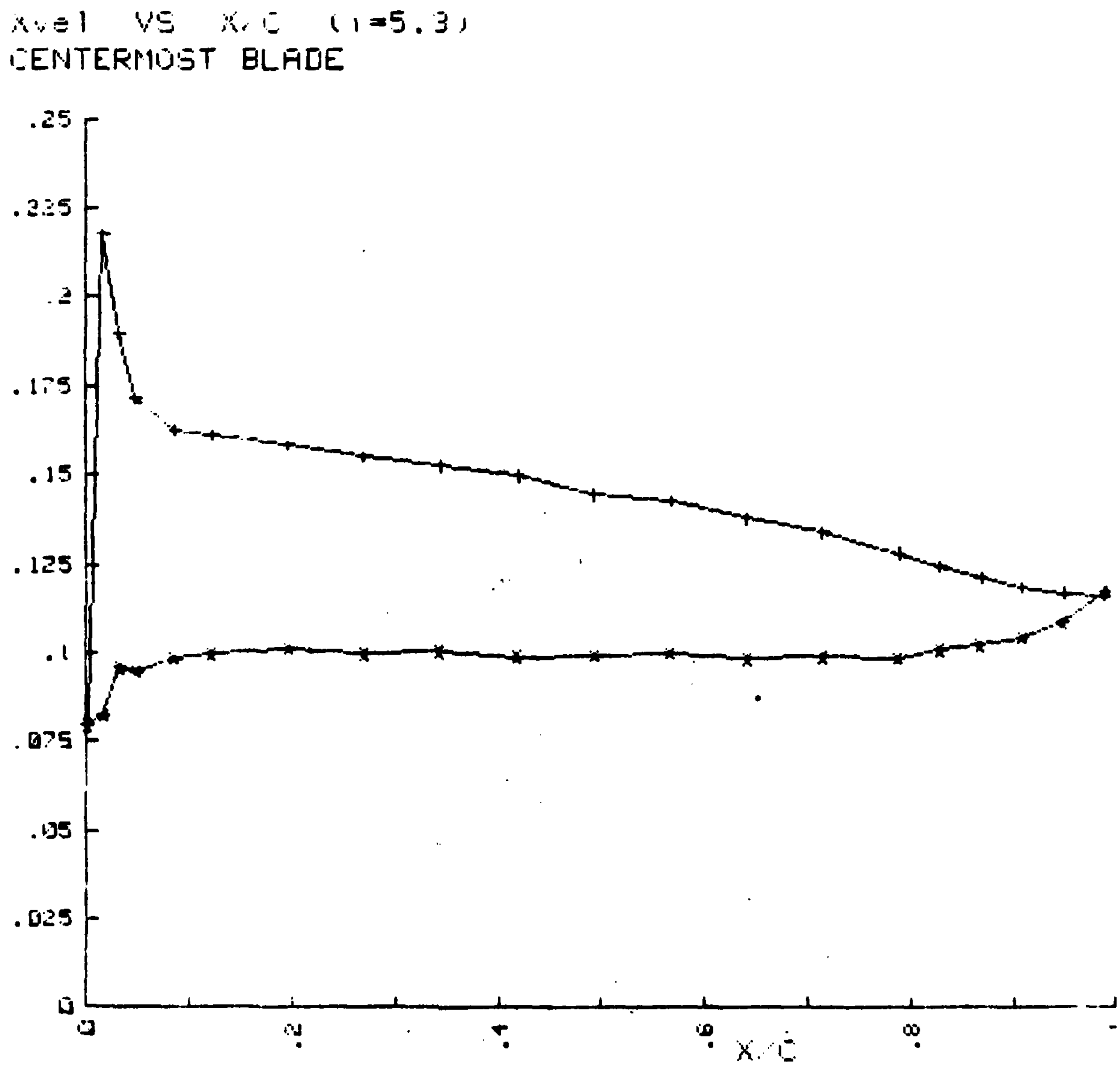


Fig. 60. Measured Blade Surface Velocity Distribution (i = 5.3, * = Pressure Side, + = Suction Side)

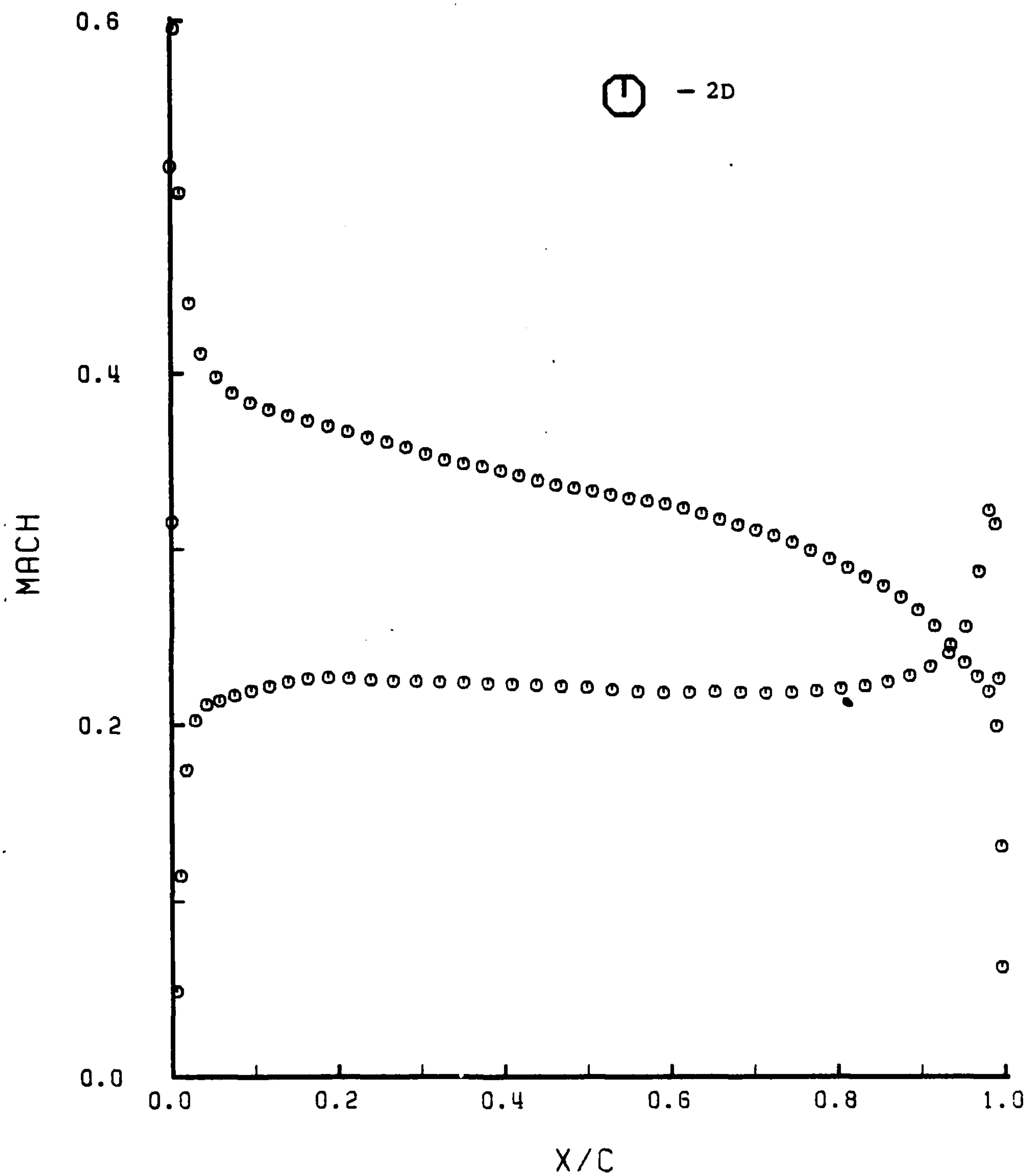


Fig. 61. 2D Code Blade Surface Mach Number Distribution (i = 5.3)

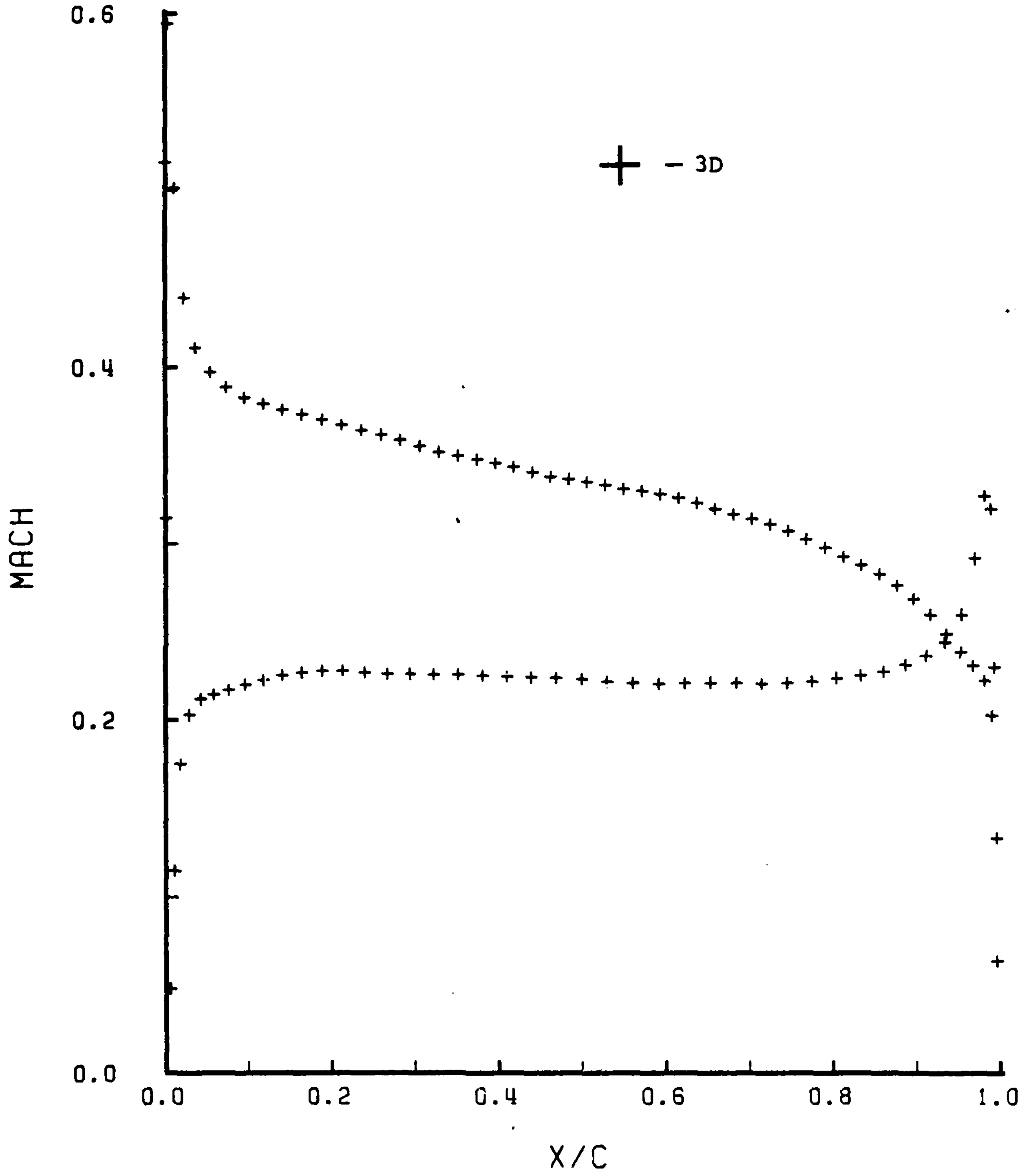


Fig. 62. 3D Code Blade Surface Mach Number Distribution (i = 5.3)

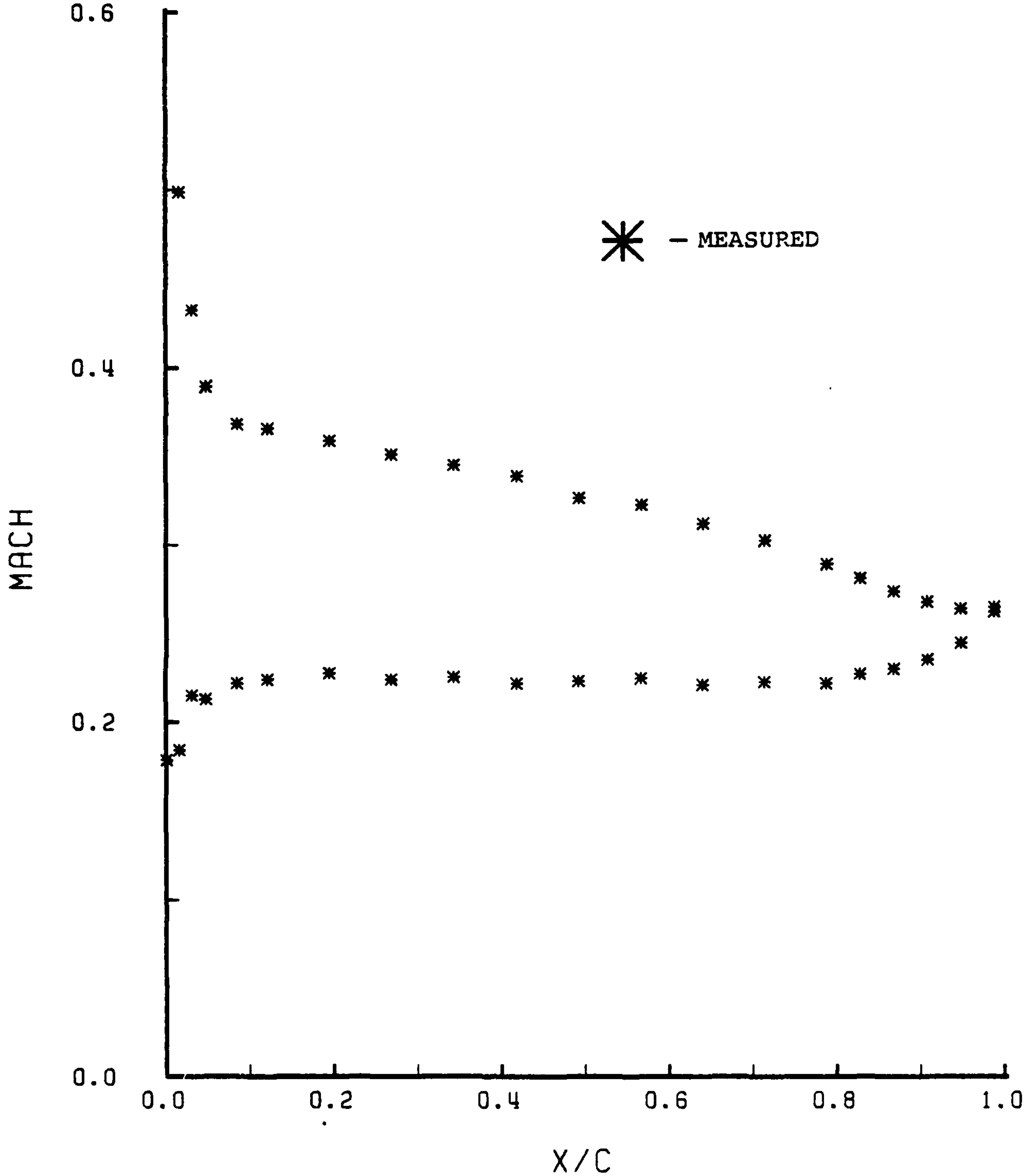


Fig. 63. Measured Blade Surface Mach Number Distribution (i = 5.3)

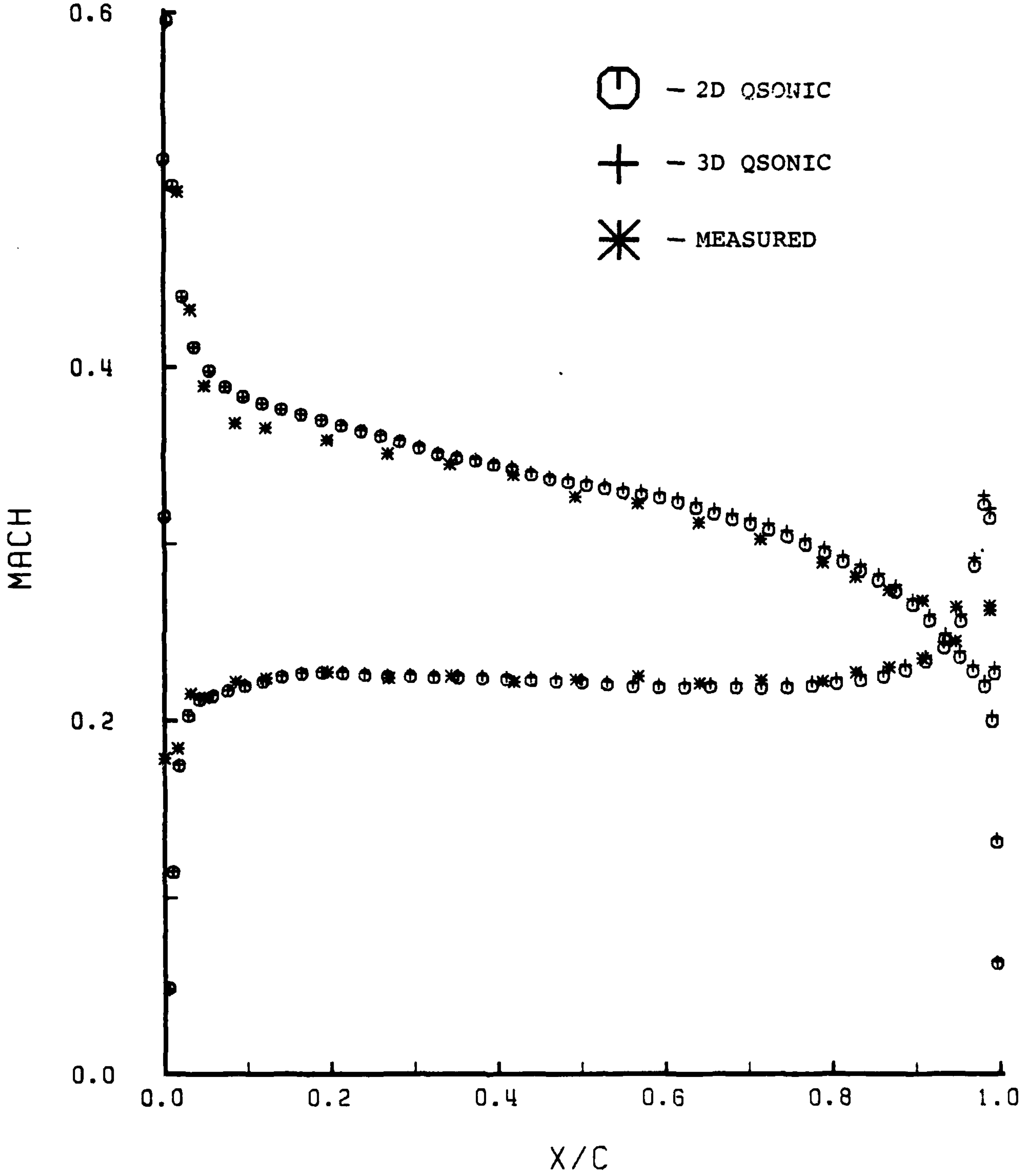


Fig. 64. Blade Surface Mach Number Distribution (i = 5.3)

APPENDIX A

MODIFICATION TO THE INLET GUIDE VANE SECTION OF THE SUBSONIC CASCADE WIND TUNNEL

As discussed in Section I, Cina discovered during his test program that while the inlet flow to the test section was uniform in direction and uniform in wall static pressure, it contained a variation in velocity and stagnation pressure resulting from the wakes of the IGV's. Because the inlet guide vanes were spaced at intervals of two inches and the test cascade blades spaced at three inches, departures from strictly periodic conditions were detected from one test blade passage to another.

To alleviate this problem, the inlet guide vane arrangement was modified so that the guide vanes were placed at 1 inch intervals. In order to preserve the option of reverting to a two inch IGV arrangement and because it was not possible to machine the south wall to hold additional blades, a separate structure was placed between the bell mouth contraction and the walls of the cascade. By mounting the IGV's in a separate unit which remained fixed once installed, hardware adjustments between tests associated with a change of end wall angle were greatly simplified.

The new inlet guide vane assembly was constructed using two lengths of 10 inch steel channel as shown in Fig. A.1.

One set of guide vanes was mounted on the south side of the unit at 2 inch spacings. A second set of vanes was mounted at 2 inch spacings on the north side of the unit. When the unit was assembled the guide vanes mounted from alternate sides meshed, resulting in an inlet guide vane spacing of 1 inch. The unit was provided with a single hand crank at the east end so that the vanes would be adjusted in unison. Once installed the complete structure could be left in place when the cascade north wall was removed to adjust air inlet angle. The one inch vane spacing ensured that periodicity at the test section would result for any test blade spacing which was a multiple of 1 inch. Equally important, the wakes remaining at the inlet to the test cascade would be greatly reduced as a result of closer spacing.

Figure A.1 shows the details of the inlet guide vane unit, while Fig. A.2 shows the assembly in relation to the bellmouth contraction and the side walls. Figure A.3 shows the mechanism to adjust the inlet guide vanes. Figure A.4 shows a view of the IGV assembly from the north side. A view of the Cascade Wind Tunnel partially assembled (north side wall off) is shown in Fig. A.5.

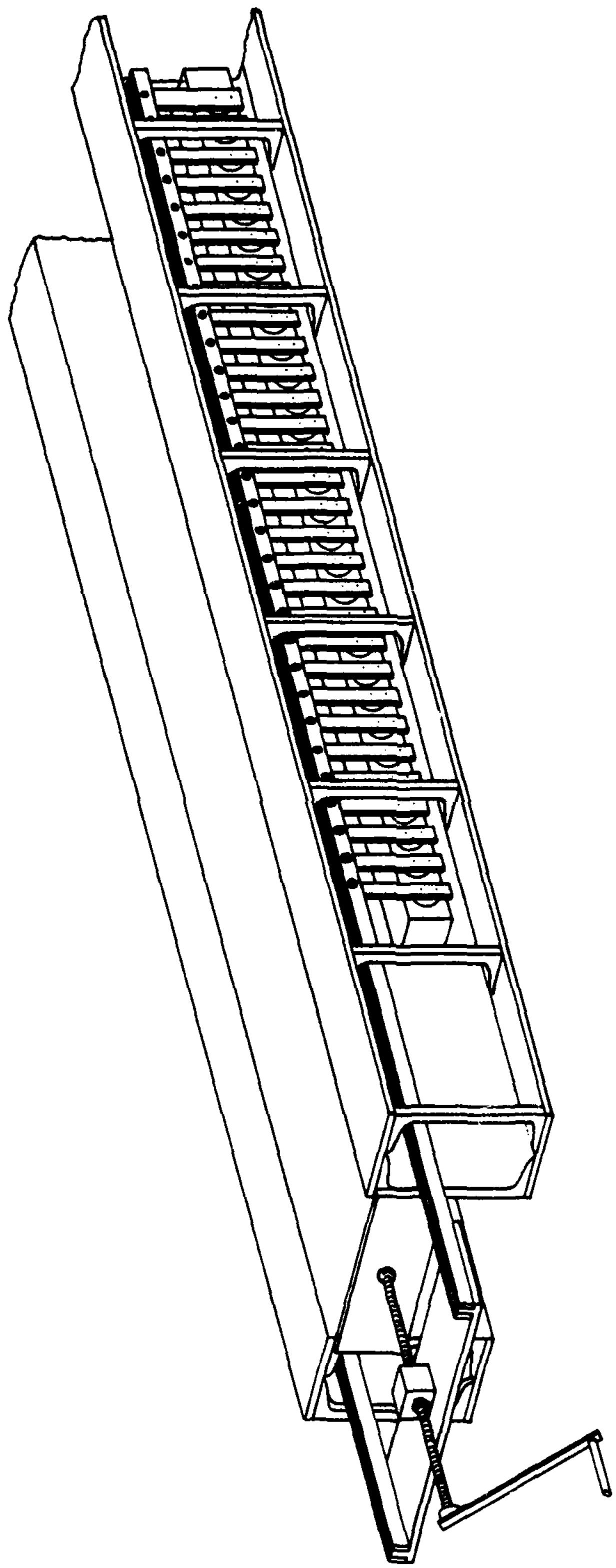


Fig. A.1. Inlet Guide Vane Assembly

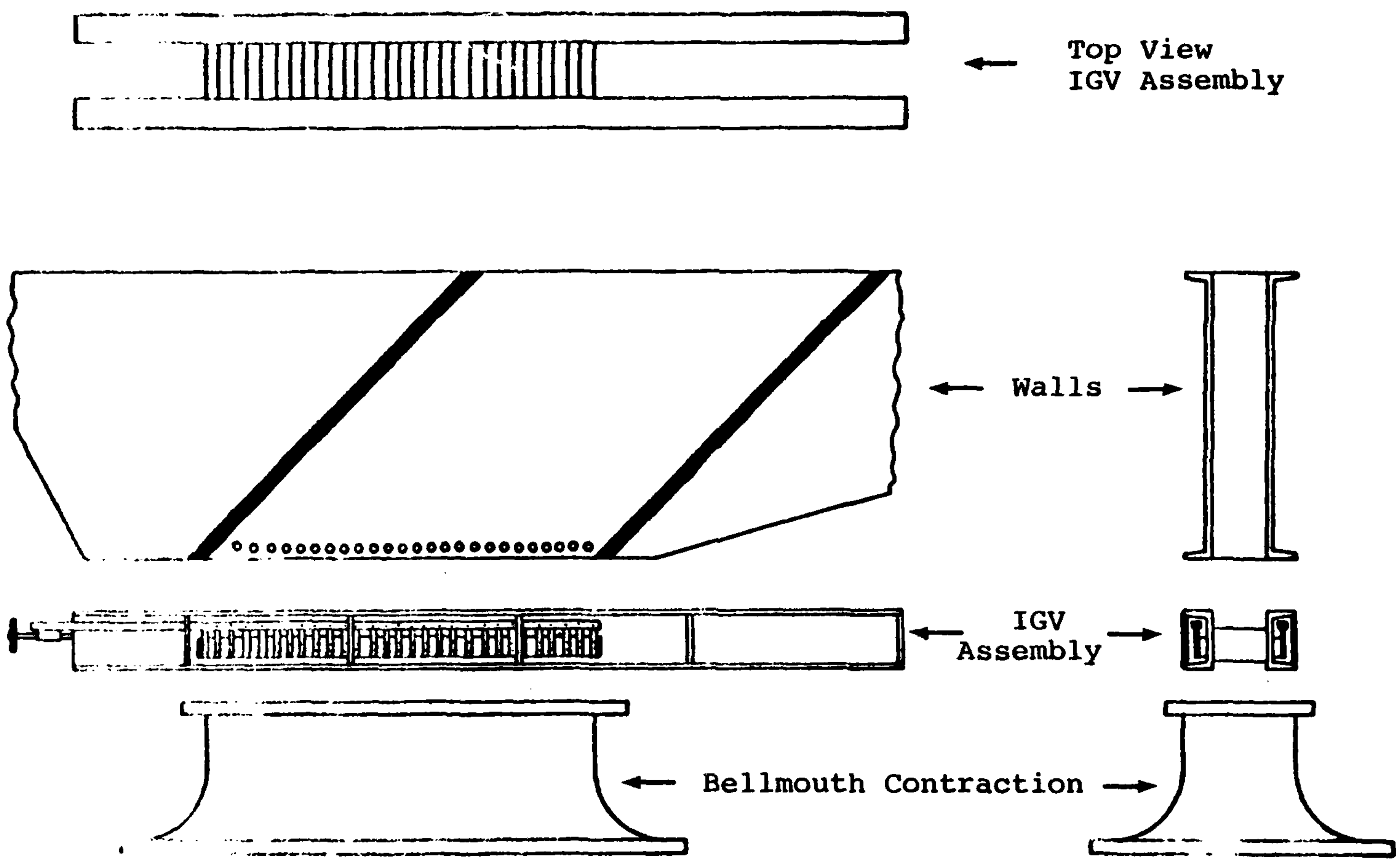


Fig. A.2. Cascade Wind Tunnel Sub-Assemblies

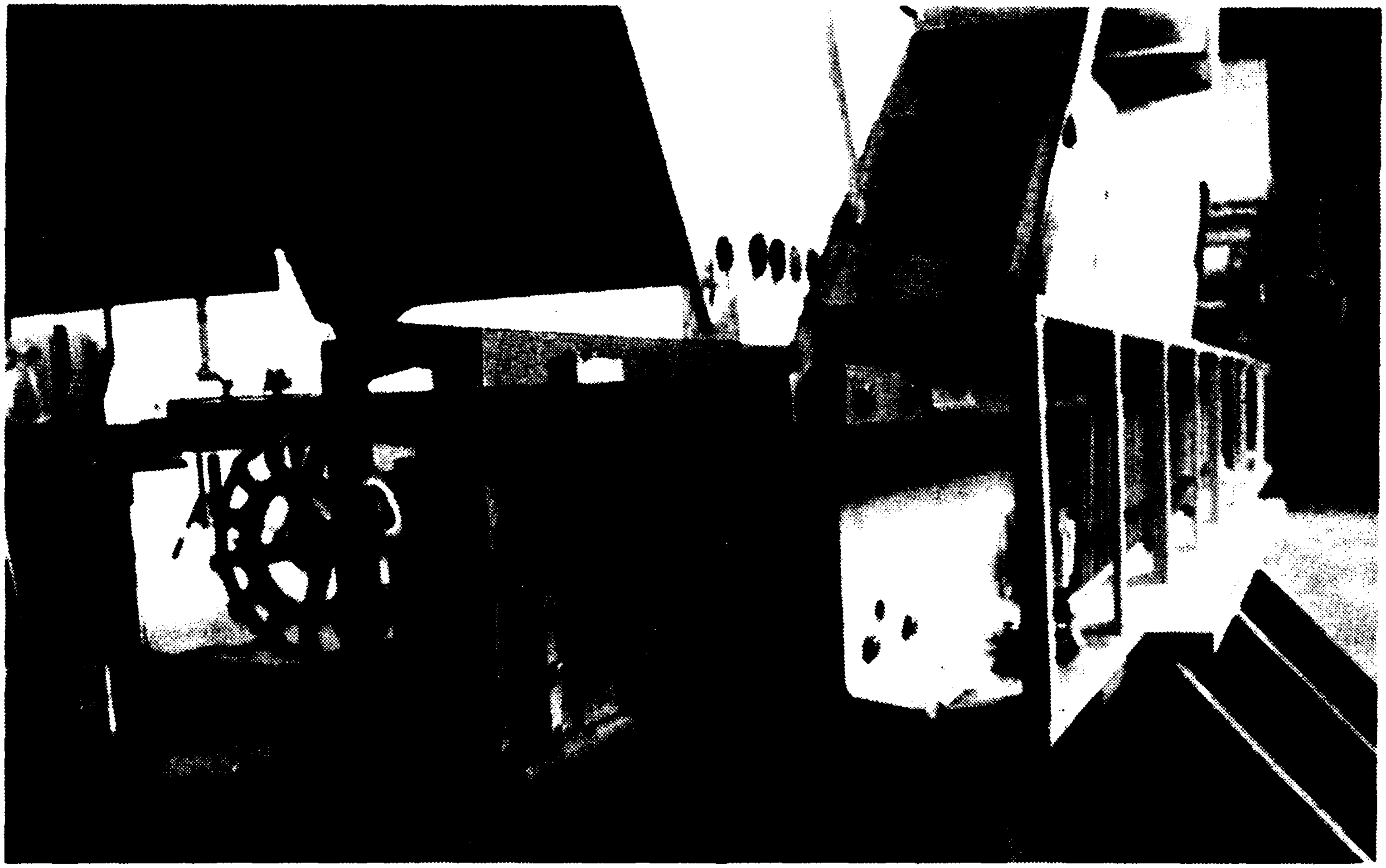


Fig. A.3. View of the IGV Adjustment Mechanism

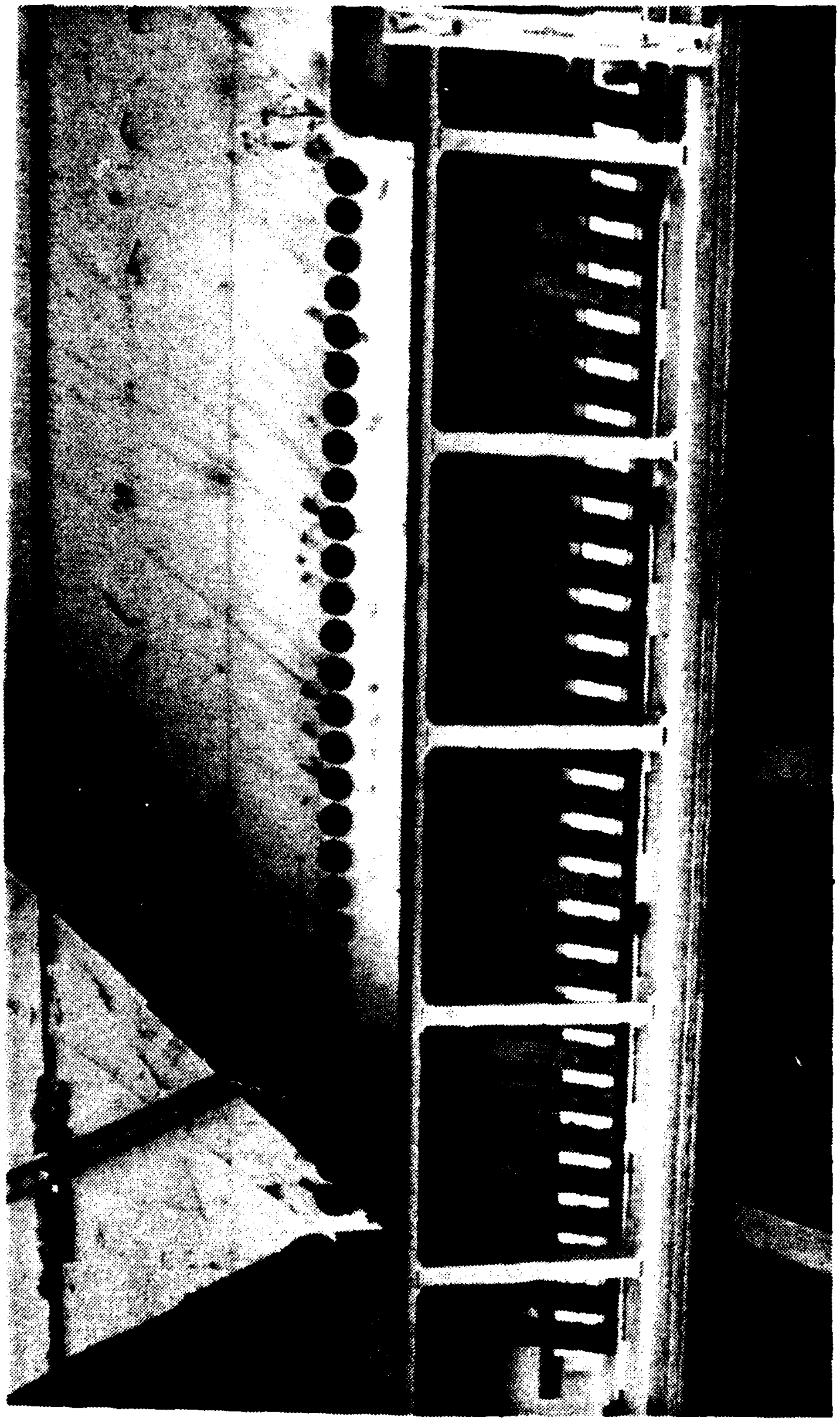


Fig. A.4. Side View of the IGV Assembly

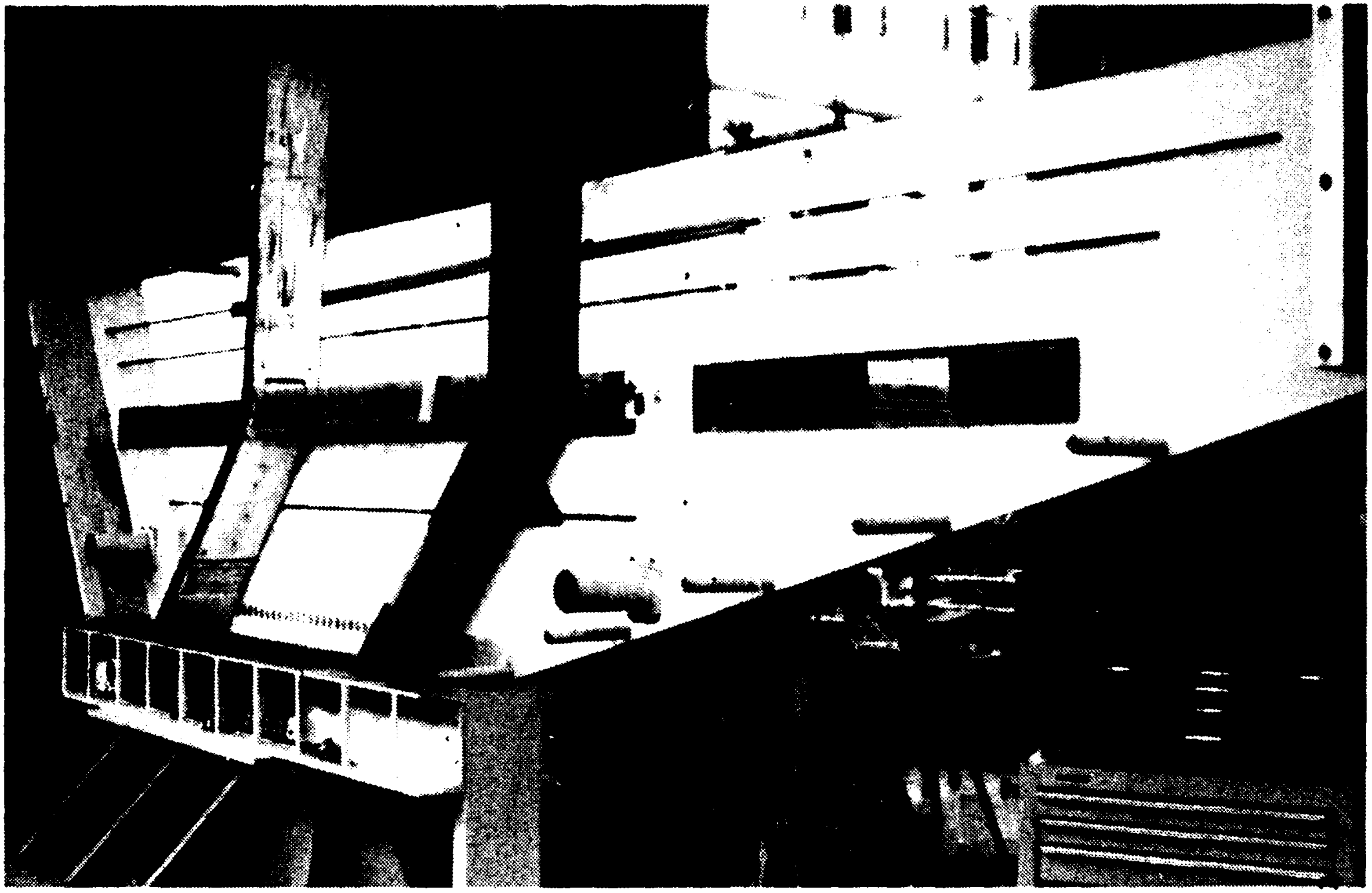


Fig. A.5. View of the Subsonic Cascade Wind Tunnel (North Wall Removed)

APPENDIX B

SELECTION AND INSTALLATION OF SCREEN MATERIAL

Pankhurst and Holder [Ref. 17] show that the turbulence of an airstream can be increased by placing a coarse mesh across the flow upstream of the test section. One of the most effective methods which is used for the reduction of turbulence and non-uniformities also consists of placing a mesh screen across the tunnel. Screens used for this purpose are of a finer mesh and are placed at a greater distance from the test section, and normally in the low-speed region upstream of the contraction in a conventional subsonic wind tunnel. By using such a screen the large scale eddies are removed at the expense of the introduction of a greater number of smaller eddies which decay rapidly.

McEligot [Ref. 16] investigated the possibilities of reducing non-uniformities in the test section of the subsonic cascade wind tunnel. His investigation and recommendations were completed while the inlet guide vanes were still at 2 inch spacings. McEligot concluded that some modification was necessary to achieve one percent uniformity for the mean velocity at the test cascade inlet plane and suggested several options. One of the options suggested was placing the turning vanes (inlet guide vanes) at a closer pitch. As explained in Appendix A, the pitch of the inlet

guide vanes was reduced from 2 inches to 1 inch. This new inlet guide vane arrangement did result in a one percent uniformity for the mean velocity at the test cascade inlet plane.

The other approach suggested was the use of wire gauze screens. McEligot showed that the velocity distribution appeared to be largely dependent on a pressure drop coefficient K, defined by the equation

$$K = \frac{p_1 - p_2}{\frac{1}{2} \rho V^2}$$

where p_1 and p_2 are the pressures upstream and downstream of the screen respectively. This pressure drop coefficient depends mainly on the blockage coefficient β defined by the equation

$$\beta = (1 - d/\ell)^2$$

where d is the diameter of wire used in the screen and ℓ is the distance between the wires. This blockage coefficient is commonly referred to as "percent open area" in catalogues of industrial wire cloth and woven wire screens.

For the velocities and flow angles used in the cascade wind tunnel, McEligot recommended using a wire gauze screen with a resistance coefficient, K, of 2.2 and a blockage coefficient, β, of 0.47. However, since the new inlet guide vane arrangement resulted in a one percent uniformity for

the mean velocity and the pressure drop across a screen with a blockage coefficient of 0.47 was expected to be higher than could be tolerated for the desired test conditions, screens with a slightly higher blockage coefficient (higher percent of open area) were selected to be tested.

The screens tested were of the following configurations:

MESH	WIRE DIAMETER (inches)	BLOCKAGE COEFFICIENT
4	.0410	.6989
5	.0410	.6320
16	.0105	.6922

Until the effectiveness of wire gauze screen in reducing non-uniformities in this facility was proven, a temporary installation of the screen material was considered adequate for testing purposes. The test screen was installed in the cascade wind tunnel by placing it between the inlet guide vane assembly and the north and south end walls. This arrangement placed the screen 7.25 inches downstream of the inlet guide vanes and 19.3 inches upstream of the lower test plane. Figure B.1 shows the installation of the wire gauze screen.

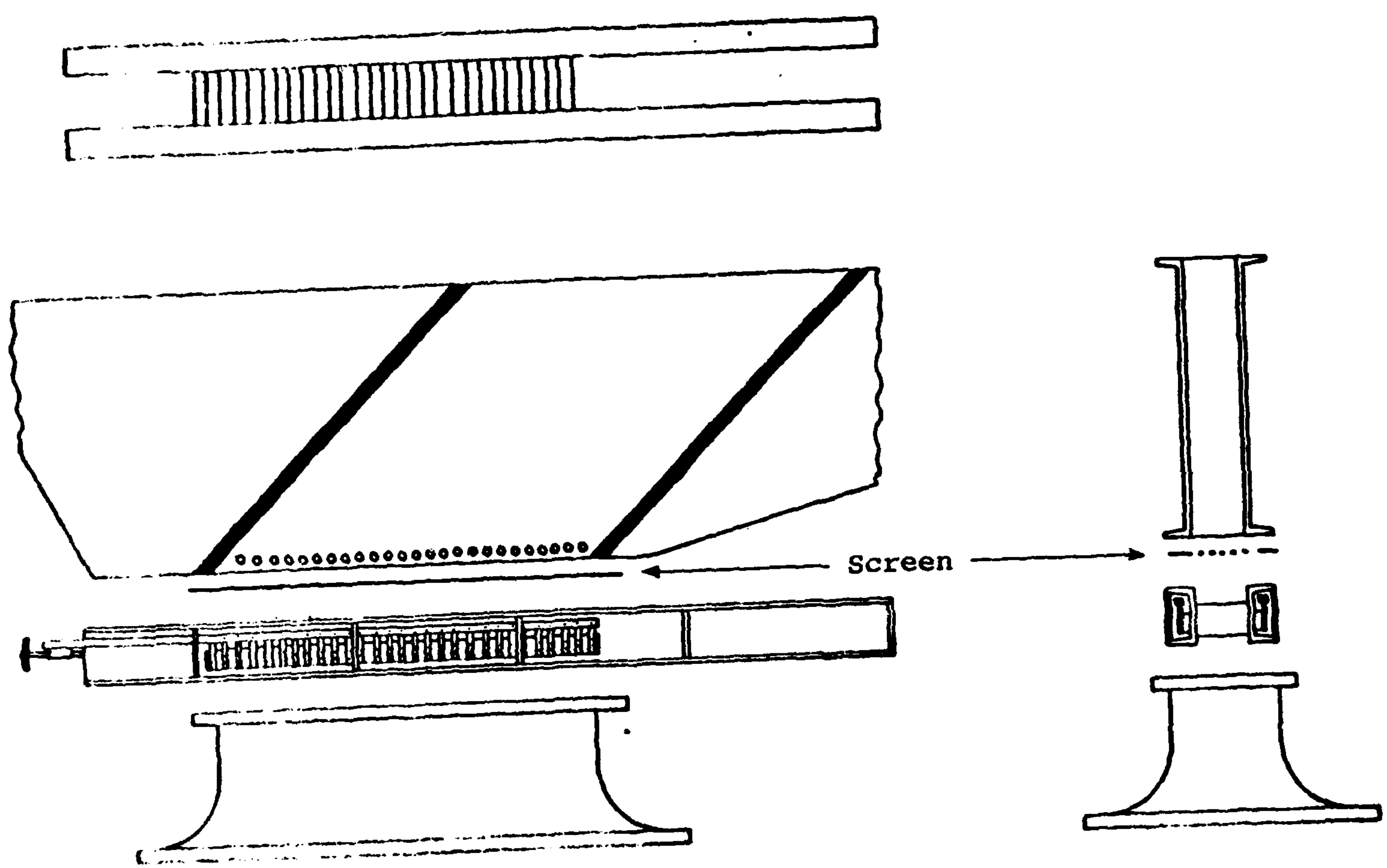


Fig. B.1. Screen Installation

APPENDIX C

CASCADE PERFORMANCE PARAMETERS

(by F. S. Cina; reproduced with minor changes from Ref. 7)

The performance of a cascade is specified in terms of the deviation angle (δ) and the loss coefficient ($\overline{\omega}$) for given inlet conditions. In Ref. 1 the loss coefficient is shown to correlate in terms of the Diffusion Factor (D). In the present work, the performance parameters were calculated using the following expressions:

1. Loss Coefficient ($\overline{\omega}$)

$$\overline{\omega} = \frac{\overline{C}_{p_{t_1}} - \overline{C}_{p_{t_2}}}{\overline{C}_{p_{t_1}} - \overline{C}_{p_1}} \qquad (1)$$

where the mass averaged pressure coefficients in Eq. (1) are defined in Appendix C of Ref. 7. It is shown in Appendix C of Ref. 7 that the effect of time dependent supply conditions are removed and the effect of AVDR is included in the use of Eq. (1).

2. Diffusion Factor (D)

$$D = 1 - \frac{W_2}{W_1} + \frac{\Delta W_u}{2\sigma W_1} \qquad (2)$$

3. <u>Pressure Rise</u>

$$C_{p_{static}} = \frac{\bar{P}_2 - \bar{P}_1}{\bar{Q}_1} \tag{3}$$

4. <u>Blade Surface Pressure Coefficients</u>

$$C_{p1} = \frac{P_s - \bar{P}_1}{\bar{Q}_1} \tag{4}$$

$$C_{p2} = \frac{P_s - \bar{P}_2}{\bar{Q}_2} \tag{5}$$

5. <u>Dimensionless Velocity</u>

$$X = \frac{V}{V_t} = \frac{V}{\sqrt{2C_pT_t}}$$

where V is the local velocity, $V_t = \sqrt{2C_pT_t}$ is the "limiting" velocity and T_t is the stagnation temperature.

APPENDIX D

INSTRUCTIONS FOR PREPARING INPUT AND OPERATING QSONIC USING A RECTILINEAR CASCADE CONFIGURATION

D.1 BACKGROUND INFORMATION

QSONIC has the capability to calculate an axial, mixed or radial flow field and the test cascade can be rotating or stationary. The geometry of the streamsurface can be a 2D planar cascade or axisymmetric with varying channel thickness and radial position. The capabilities of QSONIC, beyond those of previous cascade analysis methods (such as described in Ref. 8) include the ability to calculate through weak shocks with a peak relative Mach number less than 1.4, and completely around both leading and trailing edge regions of a blade profile. The blade shapes in the leading and trailing edge regions are not restricted to circular arcs. Detailed instructions for preparing input for a configuration other than an axial flow, stationary and rectilinear cascade may be found in Ref. 9. What follows are instructions for preparing the input applicable to the Rectilinear Cascade facility and running QSONIC on the Naval Postgraduate School's computer and associated operating system. It is assumed that the reader has a working knowledge of the NPS computer operating system and is familiar with Refs. 13 and 14.

QSONIC operates in two parts. The first part generates a body (blade) centered mesh (geometry generation). The second part actually solves for the flow conditions at points in the mesh (flow solution). The data necessary to generate a mesh consists of a two-dimensional description of the blade shape. This is in the form of pairs of (X,Y) points on the surface, together with parameters that describe the cascade layout, such as chord and stagger angle. Additionally, parameters describing the density of mesh lines complete the input for the geometry generation.

For flow field calculations, the upstream flow conditions, convergence criteria and a schedule of meshes to be used should be input. If quasi-three dimensional effects are to be considered, a data file containing a description of the streamchannel's radial thickness and position as a function of distance along the stream surface is needed. For the case described herein, this was input by assuming a linear reduction in streamchannel thickness using a factor of 1/AVDR. This gave excellent results. (The output of another NASA code, Meridl [Ref. 15], can be used to input data to QSONIC for compressor flow field calculations. This program has recently become operational on the NPS computer.)

The output of QSONIC consists of listings which contain an echo print of the input data, generated mesh coordinates on the blade surface, progress reports on the flow convergence and a list of the final velocities, pressures

and densities on the blade surface for each grid that was included in the schedule of solution meshes.

D.2 INPUT DESCRIPTION

The input for QSONIC falls into the following categories:

- Logical and case control parameters (NAMELIST PARAMS)
- Bulk data input:
  - For geometry generation runs (NAMELIST INSTUFF)
  - For flow solution runs:
    - Mesh point storage files
    - Streamchannel data file (for quasi-3D)

The following is a description of the logical and case control parameters for the Rectilinear Cascade. Except for TITLE, the format for all these variables is in namelist form. The namelist is PARAMS. This information is taken from Ref. 9 and adapted to the Cascade Wind Tunnel. Namelist variables can be entered in any order. If a default value is listed, it is not necessary to enter that particular variable. If the default value is listed as none, then a value for that parameter must be input.

The following parameters apply to both the mesh generation and the flow solution runs, but the values need not be the same.

Variable Name	Type	Default Value	Description
TITLE	Alphameric	None	This is a one-line name for the case being run. TITLE must appear on the first line of the data file that will be referred to as NAMELIST DATA.
NOFLOW	Logical	False	NOFLOW = .TRUE. if a run is to stop after generating a mesh, such as the mesh generation run. NOFLOW = .FALSE. for the flow solution run.
MS	Integer	None	MS is an array (max. dimension 10) of values (max. value = 25) for the number of grid lines in the mesh that will enclose the blade. 25 is a satisfactory number for blades of solidity near unity. As solidity increases the maximum value in MS should decrease.
NOZES	Integer	None	NOZES is an array (max. dimension 10) of values (max. value = 49) for the number of grid lines in the mesh radiating from the blade on one surface. If a value of NOZES is greater than zero, a mesh with that many lines will be developed and stored in the file MESHGEN DATA. If the value is negative, QSONIC assumes that this MESH already exists in the file MESHGEN DATA and will be read in. For Geometry Generation runs NOZES > 0 and for flow solution runs NOZES < 0. For electrostatic analog grid generator, NOZES must be odd.

BETA1	Real	None	BETA1 is the flow angle at the upstream boundary, in degrees. BETA1 is measured from the aerodynamic chordline to the direction of flow; clockwise is negative. This can be obtained from the output of the cascade tunnel data reduction program 'CX4431'. It is listed as "inlet air angle." Because of a difference in definition, it is necessary to use the usual inlet air angle minus the stagger angle for BETA1. Example: The output of CX4431 lists an inlet air angle of 42.429°. The cascade is configured with a stagger angle of 14.27°. 42.429° - 14.27 28.16 = BETA1
BETA2	Real	None	BETA2 is the flow angle at the downstream boundary in degrees. It is measured from the aerodynamic chordline to direction of flow. Clockwise is negative. This can be obtained from the output of the cascade tunnel data reduction program 'CX4431'. It is listed as outlet air angle. Because of a difference in definition, it is necessary to use the outlet air angle minus the stagger angle for BETA2.
GAMMA	Real	1.4	This is the ratio of specific heats. For the Subsonic Cascade Wind Tunnel the default value works well.
TOLS	Real	None	This is an array of dimension 10 of tolerances corresponding to MS and NOZES. Each grid solution will proceed until its TOLS value is satisfied.

MESH1	Integer	1	This is an index in the arrays MS and NOZES of the first mesh to be generated and/or used for the flow solution. For geometry generation runs, MESH1 selects one of the grids to be stored, provided NOZES(MESH1) > 0.
MESHN	Integer	None	This is an index in the arrays MS and NOZES of last grid to be calculated for geometry generation runs or used for the flow solution. For geometry generation runs, MESHN = MESH1. Subsequent flow solution runs then solve the case for all grids listed in MS and NOZES between index MESH1 and MESHN. For flow solution MESH1 normally points to the coarsest mesh. To insure equal spacing of grid lines over the entire mesh, $\frac{NOZES(I) - 1}{K(I - MESH1)} = \text{Integer},$ where MESH1 $\leq$ I $\leq$ MESHN and K = 2, 3, 4, . . . (K = 2 if grid lines are doubled between successive grids).
LAMDA0	Real	None	Stagger angle of the blade row in degrees measured from the throughflow direction to the blade chord line. (Clockwise is negative.)
CHORD	Real	None	True chord of the blade, in the same units as the blade coordinates.
S	Real	None	Blade spacing in the same units as the blade coordinates.

The following parameters are required only for the geometry generation run. QSONIC gives the user a choice of two grid (mesh) generators. The Electrostatic Analog grid generator is applicable to any blade shape and any value of stagger or turning. The interpolation scheme grid generator works for most blades except those with sharp leading edges where the edge radius is less than .5% of chord. The interpolation scheme allows the user to concentrate grid lines in areas of high interest around the blade. With the Electrostatic Analog grid generator no concentration of grid lines is available. For the cascade configuration used in this study, the interpolation scheme provided gross errors in the flow solutions, so the Electrostatic Analog grid generator was used with good results.

Name	Type	Default	Description
NED	Integer	None	Total number of body definition coordinates that are input. These are the X,Y points that describe the profile of the blade, with the first point repeated as the last point.
KN	Integer	None	KN is used to indicate which grid generator is to be used. KN = 0 will call the electrostatic analog grid generator. For interpolation scheme, KN = number of body points on upper surface from the minimum to maximum X points, inclusive.

NED and KN are required for either grid generator. If the electrostatic analog grid generator is used, no other

parameters are used. If this grid generator is used the value from NOZES that is used must be odd.

The following parameters are used only if the Interpolation Scheme is used to generate the mesh.

Name	Type	Default	Description
RLE	Real	None	RLE is the leading edge radius of the blade, with units the same as the coordinates defining the blade profile.
RTE	Real	None	RTE is the trailing edge radius of blade.
THETL	Real	0	THETL is the camber angle of the leading edge in degrees. It is measured from the aerodynamic chord to the line tangent to the mean camber line at the leading edge; clockwise is positive.
THETT	Real	0	THETT is the camber angle at the trailing edge in degrees. It is measured from the chord line to the mean camber line at the trailing edge. Clockwise is positive.
CAMPER	Integer	6	For blades whose chordline lies outside the blade profile, such as the DCA blading discussed in this report, extra grid lines surrounding the blade are needed to interpolate the blade position. The truncated value of MS( )/CAMPER is added to MS( ). The maximum allowed value of MS( ) + MS( )/CAMPER is 30 grid lines. These are the grid lines that enclose the blade profile.
STABAC	Real	0.999	STABAC is used only for test cases where no blade shape is to be input. Default value is usually adequate.

CHOP	Real	0.99	If leading edge or trailing edge radius is less than 2% of chord consult Ref. 9 for clarification. Normally, 0.9 < CHOP < 1.0.
SMOOTH	Logical	False	For automatic addition of more blade definition points in the region of the leading and trailing edges set SMOOTH = .TRUE. This should be done for all blades except cusps and wedges.
LEONLY	Logical	False	LEONLY = .TRUE. for smoothing about the leading edge only; this is used if the trailing edge is a cusp or a wedge. SMOOTH must also be .TRUE.
SLP1 SLP2 SLP3 SLP4	Real Real Real Real	1 2 1 1	These parameters control the concentration of grid lines, if desired. The default values worked well for the DCA blades reported herein. For controlling the amount and location of the concentration consult Ref. 9.

The following logical and case control parameters are required only for flow solution runs.

Name	Type	Default	Description
MINF	Real	None	This is the Mach number at the upstream boundary. This can be determined from the non-dimensional velocity X output from the cascade tunnel data reduction program 'Redd 5' and the relationship, $\frac{\gamma - 1}{2} M^2 = \frac{X^2}{1 - X^2}$ which yields $M = \left\{\frac{X^2}{1 - X^2} \cdot \frac{2}{\gamma - 1}\right\}^{1/2}$

TOLS(I) is the tolerance for MS(I) and NOZES(I). There are three forms of input permitted.

A) $-1.0 < \text{TOLS(I)} < 0.0$: Calculations of the flow solution will proceed until the relative circulation error

$$\left| \frac{C^{N}_{CALC} - C_{EXACT}}{C_{EXACT}} \right| < |\text{TOLS(I)}|$$

TOLS values between -10^{-3} and -10^{-6} are typical values for grids. This method of input is appropriate only for lifting (non-symmetric blades) cases.

B) $0.0 < \text{TOLS(I)} < 1.0$: Calculations of the flow solution will proceed until the average relative change in potential is less than the absolute value of TOLS(I).

$$\left(\frac{\delta\phi}{\phi}\right)_{AVE} < |\text{TOLS(I)}|$$

Typical values should be between 10^{-3} and 10^{-5}.

C) $1.0 < \text{TOLS(I)}$: Calculations proceed until the number of iterations equals TOLS(I).

Regardless of TOLS input, the solution for each grid will stop after 300 iterations if the TOLS criteria has not yet been made. All three forms were used for the cascade configuration reported herein with no discernible differences in results.

OVEREL	Real	1.5	The default values of these parameters are adequate for flow conditions in the subsonic cascade wind tunnel.
UNDERL	Real	1.0	
SUPREL	Real	1.0	
NOWREL	Integer	20	
NOTYET	Integer	2	
TEGARD	Real	2.0	
DAMP	Real	1.0	
CII	Real	0.2	
IT	Integer	10	Number of iterations between intermediate printouts of residuals and Mach number. The information controlled by this parameter is of limited value in comparing with measured data, so a value greater than 10 reduces the amount of computer printout. For the study reported herein 40 was used.
ALLOUT	Logical	.FALSE.	To list the flow quantities at all grid points in the last mesh set ALLOUT = .TRUE.. Unless a very coarse grid is used, the output resulting from ALLOUT = .TRUE. would be extremely voluminous and of limited value. Until the cascade is configured so it is possible to take data from between the blades, ALLOUT should be .FALSE..
QUASI3	Logical	.FALSE.	QUASI3 = .TRUE. to activate streamchannel thickness and/or radius variations. The cascade wind tunnel has no radius variations, but to simulate 3-D effects the streamchannel thickness is reduced at the exit boundary by a factor of 1/AVDR. This data is placed in a file used by QSONIC if a quasi 3-D solution is desired.
NSTRM	Integer	1	This is the position of desired streamsurface data on the streamchannel file used if QUASI3 = .TRUE.. Currently the default value of 1 identifies the proper streamsurface data

			in the streamchannel file. If the output from the NASA code 'Meridl' is used for the streamchannel data, then by using different values of NSTRM, different streamsurface data may be used.
RINF	Real	1.0	This is the spanwise radius at the upstream boundary divided by aerodynamic chord. Radius effects are activated if RINF $\neq$ 1.0. The current version of QSONIC allows the following cases.

	QUASI	RINF	Results
1)	.FALSE.	1.0	Planar 2D flow
2)	.TRUE.	$\neq$ 1.0	Thickness on file; radius on file.
3)	.TRUE.	1.0	Thickness on file; constant radius.

Only 1) and 3) apply to the cascade wind tunnel.

WAKE	Real	0.0	These parameters apply only if the test cascade is rotating and/or the downstream Mach is near 1.0.
MINF2	Real	10.0	
OMEGA	Real	0.0	
VAXIAL	Real	999.0	
FLOCO	Real	999.0	

At this point all of the logical and case control parameters necessary to use QSONIC for the flow conditions possible in the subsonic cascade wind tunnel have been discussed. The following is a description of the Bulk Data input for the geometry generation run and the flow solution run.

The format for all variables in the bulk data for mesh generation (geometry) is namelist form. The namelist is INSTUFF.

Name	Type	Default	Description
H2	Complex	None	This is a table of points defining the blade profile. The real part = X, and imaginary part = Y coordinate. The table begins at the point of maximum X value at the trailing edge and proceeds clockwise back around to the first point, which is repeated. The blade must be at the stagger angle and the origin at the point of minimum X. For Electrostatic Analog grids, the stagger angle must be positive (leading edge low, trailing edge high). The maximum number of points in H2 is 99 for the interpolation scheme or 63 for the Electrostatic Analog.
BUG2	Logical	.FALSE.	BUG2 = .TRUE. for a more detailed output of geometry generation. This will include the X,Y coordinates that define the mesh as well as second derivatives at grid points on the body. Except for trouble shooting this data is of limited value at the present time since there is no graphic output.

The bulk data for flow solutions consists of a mesh file and the streamchannel data file if quasi-3D effects are to be calculated. The mesh file is created by QSONIC during the mesh (grid) generation run. No further inputs are required from the user for the mesh file.

The streamchannel data file must contain a table of streamtube thicknesses, radial positions and corresponding X values along the streamsurface.

Name	Type	Default	Description
CHO	Real	None	CHO is the aerodynamic chord multiplied by the cosine of LAMDA0. (LAMDA0 ≡ stagger angle)
NRSP	Integer	None	NRSP is the total number of data points in each of the tables of thickness, radial position, and X location. If NRSP is 2, a linear distribution is obtained between the endpoints given. NRSP = 2 was used for the study reported herein with good results.
RM	Real	None	Array of corresponding X locations for thickness and radius data values. X = 0 represents the leading edge of blade, with the blade at stagger angle. The units can be any consistent length scale common to CHO, RM, RMSP and BESP. Inches were used in this study.
RMSP	Real	None	Spanwise radial positions of streamsurface at the X locations given in RM. RMSP was not used in the current study.
BESP	Real	None	Array of streamtube thickness values at the X locations specified in RM. For the study reported herein, at X = 0 a streamtube thickness of 1.0 was arbitrarily selected. The streamtube contraction through the test section was similated by reducing the thickness at X = 0, by a factor of 1/AVDR at the trailing edge. Duval [Ref. 3] explains AVDR.

D.3 PREPARING INPUT FILES

QSONIC was originally configured to use several input/output devices while reading data and generating output. The input/output devices are listed below as used by QSONIC.

I/O Unit	Usage
2	File containing streamchannel data. This is used only if QUASI = .TRUE..
5	Standard card input; NAMELISTS PARAMS, INSTUF.
6	Standard printed output.
13	For mesh generation runs, coordinates of all mesh points are written here. For flow solutions, X, Y, velocities, pressures and MACH are recorded for graphic display. The program currently has no graphic output capability. No user action is necessary to create this file.
18	Used as temporary storage. No output is stored here. No user action is necessary in conjunction with this file.
23	Previously developed mesh coordinates are read in from this file during the flow solution. After a mesh generation run, the user must create this file and put in it the data from I/O unit 13, so that during the flow solution QSONIC can read in the mesh points.

QSONIC is presently configured to operate with the CMS system of the IBM 370 computer. This system provides a high degree of flexibility in parameter selection. With this system, all the input/output units previously mentioned are on the disk space assigned to the Turbopropulsion Laboratory (TPL). Access to QSONIC and the TPL disk space can be obtained through the Director of TPL.

The first step in using QSONIC is the creation of the data file necessary for the mesh generation run. This is done using the standard procedures of the XEDIT function of the CMS operating system. Reference 14 has specific instructions for creating new files. The filename and filetype for the data used in this study was NAMELIST GEOMD. Once the data file is opened, the necessary data is input beginning at column 2 of the virtual card. Since the variables are in namelist form, they can be input in any order.

Table D.1 is an example of the data file necessary for the mesh generation run. The TITLE must appear on the first line (FORMAT = 20A4). After the TITLE, the logical and case control parameters are input after the namelist &PARAMS. When all the case control parameters required are input the PARAMS namelist is closed with &END. On the next line of the data file the bulk data for the mesh generation run is input in namelist form, with the namelist &INSTUF. The H1 = 100*(0.0,0.0) that appears after &INSTUF on Table D.1 was used on earlier versions of QSONIC, but is not used in the present version. It should, however, appear in the data file before the H2 variables (X,Y coordinates defining the blade profile).

At this point, some discussion of the coordinates defining the blade profile is warranted. Table D.2 is a listing of the X and Y coordinates of the DCA blading used in this study. Figure D.1 is a plot of these coordinates.

Recall that the coordinates defining the blade profile for QSONIC must be for the blade at the stagger angle and minimum X at the origin. The coordinates of Table D.2 were translated and rotated using a coordinate transformation routine for the HP-67 programmable calculator. These new coordinates appear in the namelist INSTUFF on Table D.1. Figure D.2 is a plot of the translated and rotated coordinates. It is highly recommended that such a plot be made for any new blade profiles, to ensure that the original coordinates are translated and rotated properly.

The second step in using QSONIC is the creation of the data files necessary for the flow solution run. The file used for the flow solution in this report is on the TPL disk space with a filename/filetype of NAMELIST FLOWD. The simplest way to open this data file is to use the XEDIT function, as discussed in Ref. 14, to start a new file. Then input the same data as is in the data file for the mesh generation run using the XEDIT subcommand GET (filename) (filetype). The appropriate changes and additions can then be made to this file. Table D.3 is an example of the data file just discussed.

Two more data files are required for the flow solution. The data for one of these is created by the mesh generation run. The other file contains the streamchannel thickness data for implementing quasi-3D effects.

After the mesh generation run, a file with the filename/filetype MESHGEN DATA will appear on the disk. Create a new file with the filename/filetype MESHIN DATA. This is most easily done by issuing the command 'XEDIT MESHIN DATA'; then use 'GET MESHGEN DATA'. This file contains the previously developed mesh coordinates.

The streamchannel data file should have the filename/filetype of DATA5D DATA. The format for the data file is shown below.

Virtual Card	Column No.	Variable Name
1	BLANK	
2	BLANK	
3	21-30	CHO
4	BLANK	
5	36-40	NRSP
6	BLANK	
7	BLANK	
8		
9		
10		
11		
12	1-80 (8F10.5)	RM
↓ As needed		
↓	1-80 (8F10.5)	RMSP (not used in this study)
As needed		
	1-80 (8F10.5)	BESP

Table D.4 is an example of the data file used in the present study. Since NRSP = 2 was used for this study a

linear distribution is assumed for the streamtube thickness values and only 2 values of RM and BESP are required; therefore, only 1 virtual card was required for each array.

D.4 PROGRAM OUTPUT

The output generated by QSONIC for the geometry generation run includes a printed listing (I/O unit 6) and a mesh point file (I/O unit 13). The printed listing under the CMS system I/O unit 6 is normally the computer terminal unless the command 'FILEDEF 06 PRINTER' has been invoked. It is unusual for the program to run properly the first time, so initially it is helpful to have the printed listing appear at the terminal. Once the program is running properly the output should be sent to the line printer.

The flow solution run output consists of a printed listing (I/O unit 6) and a plot data save file (I/O unit 13). Once the flow solution is running properly the printed listing should be sent to the line printer.

Table D.5 is an example of the output generated by the geometry generation run. Figure D.3 is a plot of the grid output points on the blade surface, horizontal chord, produced by the mesh generation run. Figure D.4 is a plot of the grid output points with the blade at the stagger angle. Table D.6 is an example of the output generated by the flow solution.

A det[illegible]led explanation of the printed output for the program QSONIC may be found in Ref. 9.

D.5 RUNNING THE PROGRAM

The files on the TPL disk space that apply to QSONIC are listed below:

QSONIC EXEC	A1
QSONIC FORTRAN	A1
QSONIC TEXT	A1
NAMELIST GEOM	A1
NAMELIST FLOW	A1
NAMELIST GEOMD	A1
NAMELIST FLOWD	A1
DATA5 DATA	A1
DATA5D DATA	A1

QSONIC EXEC sets the input/output devices required to read and store data. QSONIC FORTRAN is the source program. To document the changes to QSONIC necessary to use the code with the IBM 370 operating system and serve as a reference for future users, a program listing is included at the end of this appendix. QSONIC TEXT is the computer executable code created when QSONIC FORTRAN is compiled. NAMELIST GEOM and NAMELIST FLOW are the data files for the geometry generation and flow generation respectively for the example in Ref. 9. DATA5 DATA is the streamchannel data required for the quasi-3D solution for the example in Ref. 9.

NAMELIST GEOMD is the data file for the geometry generation for the DCA blading used in the study reported herein. NAMELIST FLOWD is the file for the flow solution for the cascade configuration used in this study.

QSONIC expects the input data to be in a file on the TPL disk space named NAMELIST DATA. Since the first time QSONIC is run is to develop the body centered mesh, the file NAMELIST GEOMD must be renamed NAMELIST DATA, using procedures specified in Ref. 13. Because QSONIC requires large amounts of virtual memory, extra storage must be defined for the code to operate. This is accomplished by issuing the command 'DEFINE STORAGE 1504K'.

With the data file renamed and more storage defined, type 'QSONIC' to load the program. The output will appear on the terminal screen unless FILEDEF Ø6 PRINTER was invoked prior to loading the program.

After the mesh generation is complete, rename NAMELIST DATA to NAMELIST GEOMD and change NAMELIST FLOWD to NAMELIST DATA. Create a data file with the filename/filetype MESHIN DATA. The elements of this file are the same as the elements in the file MESHGEN DATA that was created by the mesh generation run. The necessary input/output files are now configured for a flow solution run. Issue the command 'QSONIC' to begin execution.

If the program output appears at the terminal it is possible to have some I/O error messages appear with the

output. This is because the write statements in QSONIC are formatted for the 132 character long line of the printer. These errors do not affect the validity of the program output.

The explanation for any error or condition message generated by QSONIC can be found in Ref. 9.

D.6 QSONIC UPDATE

Recently an improved version of QSONIC was reported by NASA Lewis Research Center [Ref. 13]. The new version requires less virtual memory and executes approximately 30% faster than the version presently in use at NPS. Also, the output appears in a different format than is described in this appendix. Reference 18 describes the most recent version of QSONIC in detail.

TABLE D.1. INPUT DATA FOR MESH GENERATION RUN OF QSONIC

```
        NASA DCA BLADES, GEOMETRY GEN  42.4,5.3
&PARAMS NOFLOW=.TRUE.,RESTAR=.FALSE.,REMESH=4,
MS=3,5,10,15,NOZES=9,13,25,49,MESH1=4,MESHN=4,LAMDA0=14.27,
CHORD=5.01,S=3.0,NED=51,KN=0,RLE=0.44,RTE=0.44,
THETL=-45.72,THETT=45.72,CAMPER=5,STABAC=0.999,CHOP=0.99,
SMOOTH=.TRUE.,LEONLY=.FALSE.
&END
&INSTUF
H1=100*(0.0,0.0)
H2= (0.48554E+01, 0.12616E+01),(0.48095E+01, 0.12066E+01)
(0.47426E+01, 0.12102E+01),(0.45112E+01, 0.12153E+01)
(0.42827E+01, 0.12170E+01),(0.40545E+01, 0.12096E+01)
(0.38285E+01, 0.11975E+01),(0.36030E+01, 0.11793E+01)
(0.33812E+01, 0.11549E+01),(0.31591E+01, 0.11232E+01)
(0.29391E+01, 0.10879E+01),(0.27205E+01, 0.10467E+01)
(0.25038E+01, 0.9978),(0.22877E+01, 0.9428),(0.20745E+01, 0.8845)
(0.18621E+01, 0.8191),(0.16499E+01, 0.7486),(0.14402E+01, 0.6726)
(0.12322E+01, 0.5898),(0.10257E+01, 0.5011),(0.8201, 0.4045)
(0.6158, 0.3071),(0.4137, 0.2010),(0.2134, 0.882E-01)
(0.656E-01, 0.0),(0.0, 0.267E-01),(0.126E-01, 0.701E-01)
(0.1801, 0.2190),(0.3701, 0.3726),(0.5633, 0.5135),(0.7588, 0.6458)
(0.9584, 0.7657),(0.11592E+01, 0.8766),(0.13630E+01, 0.9759)
(0.15693E+01, 0.10655E+01),(0.17783E+01, 0.11486E+01)
(0.19897E+01, 0.12179E+01),(0.22017E+01, 0.12811E+01)
(0.24183E+01, 0.13341E+01),(0.26362E+01, 0.13782E+01)
(0.28562E+01, 0.14135E+01),(0.30795E+01, 0.14362E+01)
(0.33055E+01, 0.14524E+01),(0.35325E+01, 0.14565E+01)
(0.37639E+01, 0.14514E+01),(0.39971E+01, 0.14354E+01)
(0.42334E+01, 0.14109E+01),(0.44717E+01, 0.13704E+01)
(0.47147E+01, 0.13197E+01),(0.48232E+01, 0.12947E+01)
(0.48554E+01, 0.12616E+01),48*(0.0, 0.0)
BUG2= F
&END
```

TABLE D.2. TEST BLADE COORDINATES

X-COORD.	Y-PRESS.	Y-SUCT.
-0.044	0.000	0.000
-0.021	-----	0.039
0.013	-0.042	-----
0.178	0.007	0.142
0.400	0.067	0.244
0.622	0.120	0.333
0.844	0.164	0.413
1.067	0.207	0.480
1.289	0.242	0.538
1.511	0.271	0.584
1.733	0.293	0.620
1.956	0.309	0.649
2.178	0.320	0.664
2.399	0.324	0.673
2.622	0.324	0.671
2.844	0.318	0.660
3.066	0.304	0.640
3.288	0.284	0.607
3.511	0.260	0.567
3.732	0.229	0.515
3.955	0.191	0.453
4.177	0.147	0.380
4.400	0.098	0.298
4.621	0.040	0.200
4.844	-0.022	0.091
4.908	-0.042	-----
4.943	-----	0.040
4.966	0.000	0.000

TABLE D.3. INPUT DATA FOR FLOW SOLUTION RUN OF QSONIC

```
        NASA DCA BLADES, FLOW SOLUTION   42.4,5.3
&PARAMS NOFLOW=.FALSE.,RESTAR=.FALSE.,REMESH=4,
MS=3,5,10,15,NOZES=-9,-13,-25,-49,KN=0,NED=51,
MESH1=1,MESHN=4,LAMDA0=14.27,CHORD=5.01,
S=3.0,MINF=0.3127,BETA1=28.16,BETA2=-12.42,
GAMMA=1.4,TOLS=1.E+04,-2.E-05,7.E-05,6.E-05,-.000001,2*0.,
OVEREL=1.5,UNDERL=1.,SUPREL=1.0,NOWREL=20,NOTYET=10,
CII=0.40,IT=40,ALLOUT=.FALSE.,QUASI3=.TRUE.,NSTRM=1,
RINF=1.,WAKE=0.,MINF2=10.0,OMEGA=0.
&END
&INSTUF
H1=100*(0.0,0.0)
H2= (0.48554E+01, 0.12616E+01),(0.48095E+01, 0.12066E+01)
(0.47426E+01, 0.12102E+01),(0.45112E+01, 0.12153E+01)
(0.42827E+01, 0.12170E+01),(0.40545E+01, 0.12096E+01)
(0.38285E+01, 0.11975E+01),(0.36030E+01, 0.11793E+01)
(0.33812E+01, 0.11549E+01),(0.31591E+01, 0.11232E+01)
(0.29391E+01, 0.10879E+01),(0.27205E+01, 0.10467E+01)
(0.25038E+01, 0.9978),(0.22877E+01, 0.9428),(0.20745E+01, 0.8845)
(0.18621E+01, 0.8191),(0.16499E+01, 0.7486),(0.14402E+01, 0.6726)
(0.12322E+01, 0.5898),(0.10257E+01, 0.5011),(0.8201, 0.4045)
(0.6158, 0.3071),(0.4137, 0.2010),(0.2134, 0.882E-01)
(0.656E-01, 0.0),(0.0, 0.267E-01),(0.126E-01, 0.701E-01)
(0.1801, 0.2190),(0.3701, 0.3726),(0.5633, 0.5135),(0.7588, 0.6458)
(0.9584, 0.7657),(0.11592E+01, 0.8766),(0.13630E+01, 0.9759)
(0.15693E+01, 0.10655E+01),(0.17783E+01, 0.11486E+01)
(0.19897E+01, 0.12179E+01),(0.22017E+01, 0.12811E+01)
(0.24183E+01, 0.13341E+01),(0.26362E+01, 0.13782E+01)
(0.28562E+01, 0.14135E+01),(0.30795E+01, 0.14362E+01)
(0.33055E+01, 0.14524E+01),(0.35325E+01, 0.14565E+01)
(0.37639E+01, 0.14514E+01),(0.39971E+01, 0.14354E+01)
(0.42334E+01, 0.14109E+01),(0.44717E+01, 0.13704E+01)
(0.47147E+01, 0.13197E+01),(0.48232E+01, 0.12947E+01)
(0.48554E+01, 0.12616E+01),48*(0.0, 0.0)
BUG2= F
&END
```

TABLE D.4. DATA FILE FOR QUASI-3D SOLUTION

```
STREAMLINE DATA FOR DCA1
                    4.8554
                                   2



  0.0       4.85500
  1.00000   0.98476
```

TABLE D.5. SAMPLE OUTPUT FROM MESH GENERATION RUN

```
1 QSONIC VERSION 1.4.9.1               NASA DCA BLADES, GEOMETRY GEN  42.4,5.3
0     CASE CONTROL INPUT ECHO

 &PARAMS
 NOFLOW=T,RESTAR=F,REMESH=              4,MS=           3,          5,          10,
          0,          0,              0,NOZES=           9,          13,          25,
          0,          0,              0,MESH1=           4,MESHN=           4,LAMDA0=
   3.00000000     ,NED=          51,KN=           0,RLE=   .439999998     ,RTE=   .439
   45.7200012     ,CAMPER=            5,STABAC=  .999000013     ,CHOP=   .990000010
   2.00000000     ,SLP3=  1.00000000     ,SLP4=  1.00000000     ,MINF=  .0
 GAMMA=  1.39999962     ,TOLS=  .0          ,  .0          ,  .0
   .0          ,  .0          ,  .0          ,  .0          ,OVEREL=  1.
   1.00000000     ,NOWREL=           0,NOTYET=           2,TEGARD=  2.00000000     ,
          10,ALLOUT=F,QUASI3=F,NSTRM=           1,RINF=  1.00000000     ,WAKE=  .0
   .0          ,FLOCO=  999.000000     ,MINF2=  10.0000000
 &END
1                                             ELECTROSTATIC ANALOG GRID GENERATED
      NEGRID=  15 NZGRID,ODD=  49 IMAXY=  20
             SURFACE COORDINATES
             0.0130938    0.0
             0.0          0.0053293
             0.0025150    0.0139920
             0.0359481    0.0437126
             0.0738722    0.0743712
             0.1124351    0.1024950
             0.1514571    0.1289021
             0.1912974    0.1528343
             0.2313771    0.1749700
             0.2720558    0.1947904
             0.3132334    0.2126747
             0.3549501    0.2292613
             0.3971457    0.2430938
             0.4394611    0.2557086
             0.4826945    0.2662873
             0.5261875    0.2750897
             0.5700998    0.2821357
             0.6146706    0.2866666
             0.6597804    0.2899002
             0.7050898    0.2907186
             0.7512773    0.2897005
             0.7978243    0.2865070
             0.8449900    0.2816167
             0.8925548    0.2735330
             0.9410577    0.2634132
             0.9627146    0.2584230
             0.9691417    0.2518162
             0.9599799    0.2408383
             0.9466268    0.2415569
```

TABLE D.5 (Continued)

0.9004390	0.2425748
0.8548302	0.2429141
0.8092813	0.2414372
0.7641715	0.2390220
0.7191616	0.2353892
0.6748901	0.2305188
0.6305587	0.2241917
0.5866467	0.2171457
0.5430139	0.2089220
0.4997605	0.1991616
0.4566267	0.1881836
0.4140718	0.1765469
0.3716765	0.1634930
0.3293214	0.1494211
0.2874650	0.1342515
0.2459480	0.1177245
0.2047305	0.1000199
0.1636926	0.0807385
0.1229141	0.0612974
0.0825748	0.0401198
0.0425948	0.0176048
0.0130938	0.0

NUMBER OF BODY POINTS 51 PITCH 0.5988023 CHO 0.969
ITER COUNT FOR INTERNAL GRID POINTS= 72 ERR= 0.93749E-05
1 GRID OUTPUT POINTS ON BLADE SURFACE, CHORD AT ST

I	X	Y
0	0.46779E+00	0.13229E+00
1	0.46963E+00	0.12825E+00
2	0.46774E+00	0.12400E+00
3	0.46393E+00	0.12059E+00
4	0.45698E+00	0.11813E+00
5	0.44414E+00	0.11793E+00
6	0.42698E+00	0.11879E+00
7	0.40705E+00	0.11922E+00
8	0.38426E+00	0.11950E+00
9	0.35902E+00	0.11939E+00
10	0.33189E+00	0.11886E+00
11	0.30367E+00	0.11774E+00
12	0.27457E+00	0.11613E+00
13	0.24473E+00	0.11402E+00
14	0.21441E+00	0.11129E+00
15	0.18384E+00	0.10789E+00
16	0.15318E+00	0.10385E+00
17	0.12266E+00	0.99302E-01
18	0.92137E-01	0.94288E-01
19	0.61648E-01	0.88725E-01

TABLE D.5 (Cont'd)

20	0.31281E-01	0.82462E-01
21	0.13252E-02	0.75573E-01
22	-0.28214E-01	0.68206E-01
23	-0.57339E-01	0.60491E-01
24	-0.86180E-01	0.52330E-01
25	-0.11461E+00	0.43743E-01
26	-0.14254E+00	0.34746E-01
27	-0.17005E+00	0.25421E-01
28	-0.19707E+00	0.15731E-01
29	-0.22356E+00	0.56539E-02
30	-0.24939E+00	-0.47077E-02
31	-0.27451E+00	-0.15315E-01
32	-0.29883E+00	-0.26158E-01
33	-0.32227E+00	-0.36951E-01
34	-0.34480E+00	-0.47608E-01
35	-0.36650E+00	-0.58174E-01
36	-0.38734E+00	-0.68631E-01
37	-0.40685E+00	-0.78844E-01
38	-0.42504E+00	-0.88796E-01
39	-0.44122E+00	-0.97884E-01
40	-0.45542E+00	-0.10637E+00
41	-0.46752E+00	-0.11329E+00
42	-0.47730E+00	-0.11777E+00
43	-0.48484E+00	-0.12019E+00
44	-0.49043E+00	-0.12067E+00
45	-0.49468E+00	-0.11925E+00
46	-0.49685E+00	-0.11593E+00
47	-0.49483E+00	-0.11077E+00
48	-0.48883E+00	-0.10379E+00
49	-0.47905E+00	-0.94958E-01
50	-0.46696E+00	-0.84229E-01
51	-0.45218E+00	-0.71829E-01
52	-0.43554E+00	-0.58480E-01
53	-0.41686E+00	-0.44022E-01
54	-0.39721E+00	-0.29501E-01
55	-0.37669E+00	-0.15007E-01
56	-0.35583E+00	-0.10023E-02
57	-0.33468E+00	0.12473E-01
58	-0.31339E+00	0.25319E-01
59	-0.29199E+00	0.37531E-01
60	-0.27072E+00	0.49044E-01
61	-0.24951E+00	0.59877E-01
62	-0.22830E+00	0.70028E-01
63	-0.20712E+00	0.79567E-01
64	-0.18598E+00	0.88653E-01
65	-0.16492E+00	0.97197E-01
66	-0.14400E+00	0.10509E+00
67	-0.12308E+00	0.11244E+00
68	-0.10198E+00	0.11932E+00
69	-0.80804E-01	0.12573E+00
70	-0.59691E-01	0.13173E+00
71	-0.38546E-01	0.13727E+00
72	-0.17271E-01	0.14231E+00
73	0.40976E-02	0.14690E+00
74	0.25543E-01	0.15109E+00
75	0.47074E-01	0.15483E+00
76	0.68671E-01	0.15801E+00
77	0.90393E-01	0.16068E+00
78	0.11239E+00	0.16291E+00
79	0.13459E+00	0.16470E+00
80	0.15682E+00	0.16605E+00
81	0.17914E+00	0.16693E+00
82	0.20164E+00	0.16727E+00
83	0.22428E+00	0.16709E+00
84	0.24699E+00	0.16644E+00
85	0.26976E+00	0.16530E+00
86	0.29250E+00	0.16374E+00
87	0.31517E+00	0.16172E+00
88	0.33742E+00	0.15919E+00
89	0.35926E+00	0.15617E+00
90	0.38027E+00	0.15278E+00
91	0.40063E+00	0.14895E+00
92	0.41936E+00	0.14518E+00
93	0.43656E+00	0.14176E+00
94	0.45123E+00	0.13848E+00
95	0.46149E+00	0.13541E+00
96	0.46779E+00	0.13229E+00
97	0.46963E+00	0.12825E+00
98	0.46774E+00	0.12400E+00

TABLE D.6. SAMPLE OUTPUT FROM FLOW SOLUTION RUN

```
CSCAIC VERSICN 1.4.S.1          NASA UCA BLADES, FLOW SOLUTION   42.4.5.3
    CASE CONTROL INPUT ECHO

&PARAMS                          4,MS=           3,        5,        10,       15,        0,        0,        0,
ACFLCW=F,RESTAR=F,REMESH=        0,FOZCS=        -9,       -13,      -25,      -49,       0,        0,        0,
      0,              C,         0,MESH1=        1,MESHN=            4,LAMDA0=  14.2700005  ,CHCRD=  5.01000023  ,S=
      0,              C,    51,KA=          0,RLE=   .0    ,RTE=   .0         ,THETL=   .0          ,THETT=
  3.00000000    ,NED=       6,STAHAC=  .55900013   ,CHOP=  .95000010  ,SPCCTH=F,LECNLY=F,SLP1=  1.00000000  ,SLP2=
  .0            ,CAMFER=          ,SLP4=  1.00000000  ,MINF=  .31269974      ,BETA1=  28.1600037  ,BETA2= -12.4270001
  2.00000000    ,SLP3=  1.00000000        -.20000000E-04,  .69999946E-04,  .60000058E-04, -.10000034E-14,  .0
GAMMA=  1.39999522   ,TOLS=  10000.0000      .0      ,OVEREL=  1.50000000   ,UNDFRL=  1.00000000   ,SUPREL=
  .0            .0           20,ROTVET=         10,TEGARD=  2.00000000   ,DAMP=  1.00000000   ,CII=  .35999576   ,IT=
  1.00000000    ,NUMREL=                  1,RINF=  1.00000000   ,WAKE=  .0        ,OMEGA=  .0          ,VAXIAL=
       40,ALLCUT=F,QUAS13=T,NSTRM=            ,MINF2=  10.0000000
            ,FLCCC=  959.000000
  .0
&END


CALCULATED FLOW PARAMETERS
&CCNST
CIRCLN=  .369831126   ,MINF2=  .228250086   ,R2081=  .984759986   ,R2CR1=  1.00000000   ,RINF=  1.00000000   ,RATLE=
  1.00000000
&END
```

TABLE D.6 (Continued)

RELAXATION BEGINS ON GRID OF 3 SURFACE CONTOURS INTERSECTED BY 17 RADIATING LINES

SUPERSONIC RELAXATION PARAMETER, SUPREL= [illegible]000E+01

ITERATION COUNT	AVERAGE PHI CORRECTN	RESIDUAL AT T. E.	MAX MACH ON BLADE	CALCULATED CIRCULATION	RELATIVE CIRC ERROR	SUBSONIC RELAX. FAC.	SUPERSONIC RELAX. FAC.
40	0.131169E-01	0.889290E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01
80	0.105749E-01	0.889165E-02	0.313031E+00	-0.651576E-01	-0.116714E+01	0.150000E+01	0.100000E+01
120	0.747400E-02	0.889185E-02	0.313030E+00	-0.109995E+00	-0.128216E+01	0.150000E+01	0.100000E+01
160	0.577920E-02	0.889184E-02	0.313030E+00	-0.154808E+00	-0.139711E+01	0.150000E+01	0.100000E+01
200	0.471100E-02	0.889184E-02	0.313031E+00	-0.199630E+00	-0.151209E+01	0.150000E+01	0.100000E+01
240	0.397760E-02	0.889172E-02	0.313030E+00	-0.244452E+00	-0.162707E+01	0.150000E+01	0.100000E+01
280	0.344233E-02	0.889150E-02	0.313030E+00	-0.289275E+00	-0.174205E+01	0.150000E+01	0.100000E+01
300	0.322454E-02	0.889149E-02	0.313030E+00	-0.311687E+00	-0.179954E+01	0.150000E+01	0.100000E+01

FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE 3 BY 17 MESH

RADIATING LINE NO.	X/CX	Y/CX	X	Y	MACH	STATIC PRES COEF	PHI	XVEL	YVEL	DENSITY
1	1.00000	0.26029	0.46563	0.12825	0.22825	-0.25555	1.79549	0.71344	-0.05291	1.02280
2	0.93444	0.25003	0.40705	0.11922	0.22825	0.37182	1.63616	0.72092	0.07494	1.02285
3	0.76437	0.24538	0.24473	0.11402	0.22825	0.41461	-0.67092	0.72530	0.06198	1.02278
4	0.57256	0.21888	0.06165	0.08973	0.22753	0.46929	-0.80653	0.71730	0.14047	1.02284
5	0.38799	0.17175	-0.11461	0.04374	0.23588	0.42828	-0.93523	0.72277	0.22661	1.02085
6	0.22037	0.10987	-0.27459	-0.01531	0.27133	0.23670	-1.07328	0.75929	0.34281	1.01155
7	0.08171	0.04331	-0.40685	-0.07834	0.30538	0.04467	-1.21174	0.67197	0.44070	1.00218
8	0.0	0.0	-0.48484	-0.12019	0.31270	-2.51582	-0.65550	0.55387	0.72185	0.99595
9	0.00607	0.02043	-0.47905	-0.09496	0.31270	-1.62470	-2.68583	0.50606	0.79545	0.99596
10	0.11330	0.11020	-0.37669	-0.01501	0.31270	-0.06332	3.07190	0.82486	0.56148	1.00010
11	0.24655	0.13865	-0.24552	0.05587	0.28420	0.16083	-0.77307	0.81172	0.41201	1.00785
12	0.37501	0.24372	-0.12308	0.11244	0.27508	0.21751	-0.65663	0.83539	0.28154	1.01061
13	0.51226	0.27993	0.07410	0.14690	0.29268	0.11854	-0.53463	0.91835	0.18639	1.00579
14	0.64098	0.29847	0.13459	0.16470	0.28158	0.18256	-0.41129	0.89571	0.06523	1.00891
15	0.75060	0.29911	0.26576	0.16537	0.22825	5636.39062	-0.29356	20.03418	-2.44723	0.50591
16	0.92771	0.28198	0.40063	0.14895	0.22825	5634.02344	4.32693	20.07785	-2.59297	0.45596
17	1.00000	0.26029	0.46563	0.12825	0.22825	11.59704	1.48380	0.97500	-0.07226	0.95988

END OF CALCULATION ON GRID 1 OF 4

TABLE D.6 (Continued)

RELAXATION BEGINS ON GRID OF 5 SURFACE CONTOURS INTERSECTED BY 25 RADIATING LINES
SUPERSONIC RELAXATION PARAMETER, SUPREL= 0.60000E+00

ITERATION COUNT	AVERAGE PHI CORRECTN	RESIDUAL AT T. E.	MAX MACH ON BLADE	CALCULATED CIRCULATION	RELATIVE CIRC ERROR	SUBSONIC RELAX. FAC.	SUPERSONIC RELAX. FAC.
40	0.216941E-02	-0.180530E-02	0.419742E+00	0.390200E+00	0.947207E-03	0.150000E+01	0.600000E+00
55	0.113744E-02	0.195710E-03	0.410352E+00	0.389837E+00	0.139138E-04	0.150000E+01	0.600000E+00

FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE 5 BY 25 MESH

RADIATING LINE NO.	X/CX	Y/CX	X	Y	MACH	STATIC PRES COEF	PHI	XVEL	YVEL	DENSITY
1	1.00000	0.25667	0.46563	0.12825	0.20406	0.49614	-0.08157	0.65456	0.03833	1.02414
2	0.97356	0.24596	0.44413	0.11793	0.24991	0.33310	-0.10249	0.78854	0.14505	1.01623
3	0.86530	0.24747	0.35502	0.11539	0.24523	0.38189	-0.17239	0.78715	0.01052	1.01860
4	0.76677	0.24191	0.24473	0.11402	0.22662	0.47420	-0.25808	0.72556	0.05994	1.02378
5	0.64020	0.22664	0.12267	0.09937	0.22358	0.48081	-0.34508	0.70553	0.11241	1.02379
6	0.51436	0.20204	0.00132	0.07557	0.22341	0.48956	-0.43535	0.69695	0.16360	1.02392
7	0.39414	0.16903	-0.11461	0.04374	0.22944	0.46035	-0.52030	0.70329	0.22030	1.02241
8	0.26115	0.12553	-0.22356	0.01565	0.22870	0.46369	-0.60896	0.68495	0.26586	1.02257
9	0.17280	0.08535	-0.32227	-0.03695	0.24044	0.40285	-0.68443	0.65969	0.32604	1.01962
10	0.09178	0.04190	-0.40685	-0.07884	0.21771	0.52105	-0.76353	0.66031	0.31347	1.02535
11	0.02616	0.01018	-0.46752	-0.11329	0.21547	0.48223	-0.79844	0.65583	0.22161	1.02247
12	0.0	0.0	-0.49469	-0.11925	0.38120	-0.81702	-0.81540	0.89487	0.81318	0.95573
13	0.01621	0.02519	-0.47505	-0.09496	0.48649	-1.38177	-0.77073	1.10261	1.16782	0.93149
14	0.08170	0.07801	-0.41686	-0.04402	0.41463	-0.72818	-0.65978	1.05044	0.79328	0.96414
15	0.16592	0.13660	-0.33468	0.01247	0.37555	-0.43017	-0.53546	1.01533	0.63184	0.97988
16	0.25424	0.18576	-0.24552	0.05987	0.35739	-0.24872	-0.42270	1.01963	0.50815	0.98535
17	0.34147	0.22446	-0.16492	0.09720	0.34776	-0.22238	-0.31880	1.03375	0.40310	0.98661
18	0.42519	0.25405	-0.08081	0.12573	0.33987	-0.17856	-0.22133	1.04087	0.30625	0.99125
19	0.51724	0.27651	0.00410	0.14693	0.33186	-0.12541	-0.12733	1.03644	0.21254	0.99386
20	0.60673	0.29029	0.09039	0.16068	0.32225	-0.06250	-0.03578	1.02309	0.11819	0.99697
21	0.68676	0.29677	0.17914	0.16693	0.30965	0.01679	0.05444	0.99003	0.02654	1.00282
22	0.75273	0.29509	0.26576	0.16530	0.29366	0.11413	0.14219	0.93854	-0.05506	1.00557
23	0.86555	0.28561	0.35926	0.15617	0.27294	0.23328	0.22434	0.86612	-0.12315	1.01138
24	0.96571	0.27067	0.43656	0.14176	0.22317	0.47335	0.29009	0.69089	-0.19073	1.02302
25	1.00000	0.25667	0.46563	0.12825	0.20423	0.45529	0.30926	0.65511	0.03827	1.02410

END OF CALCULATION ON GRID 2 OF 4

TABLE D.6 (Continued)

RELAXATION BEGINS ON GRID OF 15 SURFACE CONTOURS INTERSECTED BY 49 RADIATING LINES

SUPERSONIC RELAXATION PARAMETER, SUPREL= 0.50000E+00

ITERATION COUNT	AVERAGE PHI CORRECTN	RESIDUAL AT T. E.	MAX MACH ON BLADE	CALCULATED CIRCULATION	RELATIVE CIRC ERROR	SUBSONIC RELAX. FAC.	SUPERSONIC RELAX. FAC.
40	0.715658E-03	-0.160764E-03	0.382182E+00	0.389971E+00	0.358700E-03	0.150000E+01	0.500000E+00
80	0.176843E-03	-0.338110E-04	0.385213E+00	0.389848E+00	0.443406E-04	0.150000E+01	0.500000E+00
120	0.697097E-04	-0.136988E-04	0.384399E+00	0.389838E+00	0.177362E-04	0.150000E+01	0.500000E+00

FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE 15 BY 49 MESH

RADIATING LINE NO.	X/CX	Y/CX	X	Y	MACH	STATIC PRES COEF	PHI	XVEL	YVEL	DENSITY
1	1.00000	0.25760	0.46563	0.12825	0.13741	0.58914	-0.04226	0.40573	0.17518	1.02864
2	0.99435	0.24565	0.46393	0.12059	0.25638	0.19790	-0.04935	0.72465	0.38451	1.00575
3	0.97357	0.24685	0.44413	0.11753	0.28225	0.17222	-0.06991	0.90285	0.05063	1.00841
4	0.93512	0.24623	0.40705	0.11422	0.24776	0.36933	-0.09584	0.79499	-0.01584	1.01755
5	0.88532	0.24840	0.35902	0.11939	0.23182	0.45495	-0.13613	0.74132	0.00953	1.02215
6	0.82793	0.24670	0.30368	0.11774	0.22423	0.48643	-0.17641	0.71966	0.03331	1.02367
7	0.76681	0.24784	0.24473	0.11402	0.22194	0.51179	-0.21855	0.70760	0.05785	1.02441
8	0.70365	0.23648	0.18385	0.10789	0.21975	0.53707	-0.26179	0.70126	0.08149	1.02467
9	0.64025	0.22758	0.12267	0.09930	0.21987	0.53679	-0.30542	0.69802	0.10948	1.02466
10	0.57679	0.21661	0.06165	0.08873	0.22139	0.50425	-0.34921	0.69515	0.13597	1.02453
11	0.51444	0.20297	0.00132	0.07557	0.22195	0.49706	-0.39303	0.69388	0.16485	1.02419
12	0.45761	0.18734	-0.05734	0.06049	0.22405	0.48729	-0.43645	0.69414	0.19073	1.02371
13	0.39423	0.16997	-0.11461	0.04374	0.22573	0.47941	-0.47959	0.69243	0.21562	1.02333
14	0.33574	0.15197	-0.17005	0.02542	0.22797	0.47305	-0.52208	0.68854	0.24090	1.02302
15	0.28126	0.13048	-0.22356	0.00565	0.22832	0.46703	-0.56381	0.68322	0.26675	1.02273
16	0.22844	0.10874	-0.27450	-0.01531	0.22984	0.45962	-0.60433	0.67768	0.29297	1.02238
17	0.17892	0.08630	-0.32227	-0.03695	0.23184	0.45512	-0.64320	0.67199	0.31337	1.02215
18	0.13316	0.06430	-0.36651	-0.05817	0.22917	0.46166	-0.67959	0.66037	0.32744	1.02247
19	0.09122	0.04287	-0.40685	-0.07884	0.22572	0.47942	-0.71280	0.64001	0.34098	1.02333
20	0.05555	0.02312	-0.44122	-0.09788	0.22115	0.50086	-0.74101	0.61751	0.35168	1.02437
21	0.02631	0.00715	-0.46752	-0.11329	0.20281	0.57781	-0.76245	0.58437	0.28961	1.02809
22	0.01036	0.0	-0.48484	-0.12019	0.12594	0.80755	-0.77368	0.39303	0.10126	1.03518
23	0.00015	0.00041	-0.49468	-0.11925	0.46514	-2.65762	-0.77397	0.13725	-1.43851	0.86640
24	0.0	0.00976	-0.49483	-0.11177	0.34739	-3.54931	-0.75900	0.51966	-0.03267	0.81972
25	0.01636	0.02616	-0.47905	-0.09496	0.42428	-0.60051	-0.71873	0.98465	0.91761	0.96055
26	0.04422	0.05214	-0.45218	-0.07183	0.39787	-0.59262	-0.67297	0.97617	0.80448	0.97086
27	0.08034	0.07697	-0.41686	-0.04402	0.38430	-0.49952	-0.61727	0.97622	0.73668	0.97590
28	0.12245	0.10505	-0.37669	-0.01591	0.37726	-0.43992	-0.55733	0.98564	0.68081	0.97845
29	0.16615	0.13755	-0.33468	0.01247	0.37195	-0.40053	-0.49742	1.00678	0.62453	0.98034
30	0.21031	0.16353	-0.29199	0.03753	0.36673	-0.36310	-0.43916	1.02119	0.56811	0.98218
31	0.25435	0.18670	-0.24952	0.05987	0.36108	-0.32318	-0.38345	1.03156	0.51064	0.98415
32	0.29332	0.20711	-0.20711	0.07556	0.35539	-0.28306	-0.33006	1.03696	0.45741	0.98612
33	0.34207	0.22546	-0.16497	0.09729	0.35165	-0.25142	-0.27860	1.04177	0.40738	0.98773
34	0.38545	0.24120	-0.12304	0.11244	0.34583	-0.21742	-0.22907	1.04515	0.35466	0.98935
35	0.42928	0.25498	-0.08081	0.12573	0.34075	-0.18287	-0.18053	1.04368	0.30645	0.99104
36	0.47210	0.26694	-0.03854	0.13726	0.33430	-0.15327	-0.13319	1.04193	0.25954	0.99250
37	0.51751	0.2769[illegible]	0.00410	0.14690	0.33216	-0.12565	-0.08652	1.03574	0.21335	0.99385
38	0.56138	0.28515	0.04704	0.15483	0.32769	-0.09654	-0.04042	1.03384	0.16573	0.99527
39	0.60679	0.29122	0.09039	0.16068	0.32270	-0.05963	0.00502	1.02262	0.11567	0.99708
40	0.65261	0.29530	0.13459	0.16470	0.31548	-0.01804	0.05024	1.00620	0.07110	0.99912
41	0.69581	0.29769	0.17514	0.16693	0.30355	0.02521	0.09480	0.98662	0.02641	1.00123
42	0.74540	0.29787	0.22427	0.16710	0.30017	0.07666	0.13883	0.96049	-0.01634	1.00375
43	0.79276	0.29601	0.26576	0.16530	0.29045	0.13460	0.18190	0.92859	-0.05463	1.00657
44	0.83585	0.29225	0.31517	0.16172	0.27958	0.19768	0.22357	0.89119	-0.09032	1.00965
45	0.88557	0.28654	0.35926	0.15617	0.26543	0.27539	0.26256	0.84177	-0.12573	1.01348
46	0.92946	0.27905	0.40063	0.14895	0.24805	0.36360	0.29720	0.78254	-0.14620	1.01796
47	0.96571	0.27160	0.43656	0.14176	0.22901	0.46266	0.32531	0.71753	-0.16223	1.02252
48	0.99157	0.26501	0.46149	0.13541	0.17384	0.66371	0.34344	0.50833	-0.23369	1.03224
49	1.00000	0.25760	0.46563	0.12825	0.13755	0.58829	0.34758	0.40616	0.17527	1.02860

END OF CALCULATION ON GRID 3 OF 4

TABLE D.6 (Continued)

RELAXATION BEGINS ON GRID OF 15 SURFACE CONTOURS INTERSECTED BY 97 RADIATING LINES

SUPERSONIC RELAXATION PARAMETER, SUPREL= 0.50000E+00

ITERATION COUNT	AVERAGE PHI CORRECTN	RESIDUAL AT T. E.	MAX MACH ON BLADE	CALCULATED CIRCULATION	RELATIVE CIRC ERROR	SUBSONIC RELAX. FAC.	SUPERSONIC RELAX. FAC.
50	0.842054E-04	-0.341288E-04	0.381786E+00	0.389843E+00	0.314971E-04	0.150000E+01	0.500000E+00
54	0.586785E-04	-0.273761E-04	0.382888E+00	0.389840E+00	0.229348E-04	0.150000E+01	0.500000E+00

FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE 15 BY 97 MESH

RADIATING LINE NO.	X/CX	Y/CX	X	Y	MACH	STATIC PRES COEF	PHI	XVEL	YVEL	DENSITY
1	1.00000	0.25756	0.46563	0.12825	0.04384	0.82554	-0.03819	0.17131	0.11397	1.04004
2	0.99474	0.25315	0.46774	0.12400	0.23020	0.39126	-0.14746	0.46586	0.57035	1.01506
3	0.99411	0.24563	0.46393	0.12059	0.31951	-0.06778	-0.14535	0.88529	0.50842	0.95571
4	0.98692	0.24718	0.45693	0.11813	0.32722	-0.12417	-0.05356	1.02703	0.19377	0.99392
5	0.97262	0.24688	0.44413	0.11793	0.29199	0.12106	-0.06641	0.93477	-0.01479	1.00591
6	0.95528	0.24777	0.42698	0.11879	0.25982	0.30647	-0.08129	0.83281	-0.03039	1.01494
7	0.93525	0.24821	0.40705	0.11922	0.24444	0.38775	-0.09722	0.79452	-0.01326	1.01889
8	0.91167	0.24856	0.38426	0.11959	0.23665	0.42657	-0.11477	0.76002	-0.00302	1.02077
9	0.88556	0.24838	0.35902	0.11939	0.23133	0.45275	-0.13370	0.74296	0.00883	1.02204
10	0.85745	0.24763	0.33189	0.11886	0.22770	0.47011	-0.15367	0.73117	0.02158	1.02288
11	0.82629	0.24668	0.30368	0.11774	0.22540	0.48106	-0.17420	0.72335	0.03424	1.02341
12	0.79317	0.24502	0.27457	0.11613	0.22367	0.48920	-0.19523	0.71723	0.04519	1.02380
13	0.76730	0.24283	0.24473	0.11402	0.22279	0.49656	-0.21665	0.71134	0.05717	1.02416
14	0.72593	0.24011	0.21441	0.11129	0.22103	0.50151	-0.23830	0.70668	0.07106	1.02440
15	0.70431	0.23645	0.18385	0.10789	0.22177	0.50269	-0.26011	0.70475	0.08561	1.02446
16	0.67258	0.23230	0.15318	0.10385	0.22107	0.50134	-0.28207	0.70347	0.09979	1.02439
17	0.64171	0.22760	0.12267	0.09931	0.22125	0.50049	-0.30400	0.70241	0.10997	1.02435
18	0.60962	0.22241	0.09214	0.09429	0.22096	0.50187	-0.32599	0.69561	0.12125	1.02442
19	0.57787	0.21606	0.06165	0.08873	0.22065	0.50321	-0.34757	0.69603	0.13525	1.02448
20	0.54645	0.21018	0.03129	0.08246	0.22107	0.50125	-0.36956	0.69409	0.15138	1.02439
21	0.51545	0.20305	0.00132	0.07557	0.22224	0.49586	-0.39184	0.69444	0.16646	1.02413
22	0.48485	0.19443	-0.02821	0.06821	0.22340	0.49145	-0.41365	0.69520	0.17877	1.02396
23	0.45475	0.18744	-0.05734	0.06049	0.22408	0.48730	-0.43537	0.69434	0.19020	1.02371
24	0.42492	0.17510	-0.08618	0.05233	0.22450	0.48531	-0.45652	0.69230	0.20250	1.02362
25	0.39550	0.17311	-0.11461	0.04374	0.22501	0.48288	-0.47837	0.69012	0.21535	1.02350
26	0.36660	0.16241	-0.14254	0.03475	0.22552	0.48152	-0.49561	0.68795	0.22736	1.02336
27	0.33813	0.15116	-0.17005	0.02542	0.22572	0.47954	-0.52068	0.68474	0.23881	1.02334
28	0.31017	0.14113	-0.19708	0.01573	0.22583	0.47904	-0.54159	0.68054	0.25147	1.02331
29	0.28277	0.13076	-0.22356	0.00565	0.22610	0.47777	-0.56207	0.67655	0.26439	1.02325
30	0.25614	0.11598	-0.24939	-0.00471	0.22641	0.47632	-0.58230	0.67259	0.27692	1.02318
31	0.23016	0.10501	-0.27459	-0.01531	0.22696	0.47363	-0.60214	0.66883	0.29032	1.02305
32	0.20489	0.09775	-0.29882	-0.02616	0.22777	0.46983	-0.62159	0.66649	0.30159	1.02287
33	0.18064	0.08562	-0.32227	-0.03695	0.22801	0.46871	-0.64050	0.66376	0.30975	1.02281
34	0.15732	0.07555	-0.34480	-0.04761	0.22720	0.47258	-0.65875	0.65805	0.31575	1.02300
35	0.13417	0.06466	-0.36650	-0.05817	0.22544	0.48193	-0.67632	0.64929	0.32097	1.02340
36	0.11331	0.05334	-0.38733	-0.06863	0.22296	0.49251	-0.69312	0.63752	0.32677	1.02396
37	0.09312	0.04328	-0.40635	-0.07884	0.22734	0.50463	-0.70880	0.62430	0.33409	1.02455
38	0.07430	0.03298	-0.42514	-0.08881	0.21735	0.51838	-0.72340	0.61096	0.33867	1.02522
39	0.05756	0.02357	-0.44122	-0.09784	0.21451	0.53123	-0.73626	0.59658	0.34567	1.02584
40	0.04287	0.01479	-0.45542	-0.10637	0.21228	0.54128	-0.74761	0.58912	0.34418	1.02633
41	0.03034	0.00764	-0.46752	-0.11329	0.20314	0.57983	-0.75708	0.58060	0.29938	1.02819
42	0.02023	0.00390	-0.47731	-0.11777	0.17516	0.68619	-0.76383	0.52431	0.20746	1.03342
43	0.01243	0.00150	-0.48484	-0.12019	0.11493	0.86249	-0.76775	0.36020	0.09700	1.04182
44	0.00654	0.0	-0.49043	-0.12067	0.04805	0.85274	-0.76854	-0.12434	0.09262	1.04135
45	0.00224	0.00147	-0.49464	-0.11925	0.31438	-0.34320	-0.76699	-0.67528	0.74018	0.98202
46	0.0	0.00431	-0.49685	-0.11593	0.51588	-2.18975	-0.76110	-0.05546	1.61058	0.85549
47	0.00239	0.01324	-0.49683	-0.11[illegible]	0.59442	-2.30750	-0.74979	0.93068	1.60215	0.87592
48	0.00823	0.01747	-0.48633	-0.10379	0.57153	-1.46434	-0.73407	1.09724	1.17607	0.92734
49	0.01462	0.02660	-0.47505	-0.09[illegible]	0.43935	-0.91983	-0.71493	1.02719	0.97843	0.95461
50	0.02393	0.03776	-0.46696	-0.08[illegible]	0.41179	-0.65589	-0.69344	0.98744	0.85264	0.96596
51	0.04421	0.05553	-0.45218	-0.07184	0.39744	-0.58835	-0.66875	0.97669	0.80164	0.97107
52	0.06344	0.06435	-0.43554	-0.05[illegible]	0.38850	-0.52111	-0.64215	0.97080	0.76499	0.97435
53	0.08277	0.07631	-0.41786	[illegible]	0.38202	-0.47979	-0.61320	0.97195	0.73527	0.97643
54	0.10395	0.09433	-0.39721	[illegible]	0.37905	-0.45172	-0.58359	0.97819	0.70691	0.97702
55	0.12432	0.10533	-0.37669	[illegible]	0.37596	-0.42892	-0.55340	0.98597	0.67917	0.97894
56	0.14591	0.12282	-0.35583	[illegible]	0.37310	-0.40823	-0.52342	0.99446	0.65153	0.97996
57	0.16779	0.13776	-0.33468	0.01247	0.37120	-0.38736	-0.49373	1.00220	0.62165	0.98099
58	0.18952	0.15106	-0.31339	0.02532	0.36716	-0.36576	-0.46451	1.00895	0.59233	0.98206
59	0.21196	0.16369	-0.29199	0.03753	0.36417	-0.34434	-0.43581	1.01447	0.56395	0.98311
60	0.23397	0.17560	-0.27072	0.04904	0.36146	-0.32520	-0.40784	1.01581	0.53625	0.98415
61	0.25591	0.18690	-0.24952	0.05987	0.35866	-0.30558	-0.38050	1.02498	0.50700	0.98502
62	0.27736	0.19751	-0.22830	0.07003	0.35514	-0.28102	-0.35372	1.02725	0.47658	0.98622
63	0.29575	0.20718	-0.20711	0.07956	0.35172	-0.25724	-0.32754	1.02097	0.45193	0.98739
64	0.32165	0.21658	-0.18598	0.08865	0.34960	-0.24266	-0.30183	1.02910	0.43005	0.98811
65	0.34345	0.22542	-0.16492	0.09720	0.34807	-0.23224	-0.27653	1.03417	0.40477	0.98862
66	0.36535	0.23355	-0.14400	0.10539	0.34674	-0.21831	-0.25175	1.03751	0.37800	0.98930
67	0.38673	0.24120	-0.12308	0.11244	0.34351	-0.20115	-0.22736	1.03845	0.35164	0.99014

TABLE D.6 (Continued)

68	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
69	0.43147	0.25495	-0.08081	0.12573	0.33833	-0.16625	-0.17914	1.03635	0.30432	0.95196
70	0.45232	0.26115	-0.05969	0.13173	0.33675	-0.15575	-0.15550	1.03734	0.28309	0.95337
71	0.47420	0.26688	-0.03854	0.13726	0.33525	-0.14613	-0.13234	1.03895	0.25920	0.95295
72	0.49621	0.27210	-0.01727	0.14231	0.33331	-0.13288	-0.10968	1.03431	0.23472	0.95349
73	0.51832	0.27685	0.00410	0.14690	0.33127	-0.11943	-0.08549	1.03260	0.21247	0.95415
74	0.54152	0.28118	0.02555	0.15109	0.32986	-0.11076	-0.06243	1.03619	0.19117	0.95461
75	0.56279	0.28506	0.04708	0.15483	0.32843	-0.10300	-0.03944	1.03604	0.16612	0.95506
76	0.58514	0.28834	0.06867	0.15807	0.32677	-0.09552	-0.01659	1.03251	0.13945	0.95582
77	0.60761	0.29111	0.09039	0.16068	0.32301	-0.06568	0.00611	1.02584	0.11527	0.95679
78	0.63037	0.29342	0.11239	0.16291	0.31985	-0.04539	0.02883	1.01620	0.09264	0.95778
79	0.65334	0.29526	0.13459	0.16470	0.31684	-0.02626	0.05148	1.01047	0.07141	0.95872
80	0.67634	0.29666	0.15682	0.16605	0.31405	-0.00885	0.07395	1.00310	0.05029	0.95957
81	0.69944	0.29757	0.17914	0.16693	0.31106	0.01009	0.09629	0.99448	0.02709	1.00049
82	0.72272	0.29792	0.20164	0.16726	0.30722	0.03343	0.11856	0.98230	0.00381	1.00166
83	0.74613	0.29775	0.22427	0.16710	0.30275	0.06140	0.14065	0.96861	-0.01760	1.00300
84	0.76964	0.29706	0.24699	0.16644	0.29791	0.09175	0.16250	0.95283	-0.03745	1.00443
85	0.79320	0.29585	0.26976	0.16533	0.29290	0.12073	0.18406	0.93615	-0.05543	1.00590
86	0.81673	0.29427	0.29251	0.16374	0.28793	0.15003	0.20527	0.91529	-0.07252	1.00732
87	0.84019	0.29218	0.31517	0.16172	0.28267	0.18054	0.22607	0.90096	-0.09116	1.00891
88	0.86321	0.28557	0.33762	0.15919	0.27634	0.21665	0.24616	0.87863	-0.11061	1.01057
89	0.88581	0.28644	0.35926	0.15617	0.26800	0.25871	0.26543	0.85239	-0.12746	1.01262
90	0.90755	0.28294	0.38027	0.15278	0.25976	0.30774	0.28351	0.82078	-0.14311	1.01500
91	0.92851	0.27897	0.40063	0.14895	0.24895	0.36434	0.30043	0.78436	-0.15261	1.01775
92	0.94795	0.27507	0.41936	0.14518	0.23866	0.41579	0.31535	0.75209	-0.15354	1.02025
93	0.96578	0.27153	0.43656	0.14176	0.23116	0.45374	0.32855	0.72631	-0.15322	1.02209
94	0.98096	0.26814	0.45123	0.13848	0.22244	0.49405	0.33555	0.65173	-0.17992	1.02404
95	0.99158	0.26496	0.46149	0.13541	0.20247	0.57906	0.34704	0.60714	-0.23519	1.02815
96	0.99805	0.26174	0.46778	0.13229	0.13310	0.76638	0.35129	0.33378	-0.26925	1.03816
97	1.00000	0.25156	0.46963	0.12825	0.06403	0.82445	0.35166	0.17175	0.11443	1.03599

END OF CALCULATICN CN GRID 4 CF 4

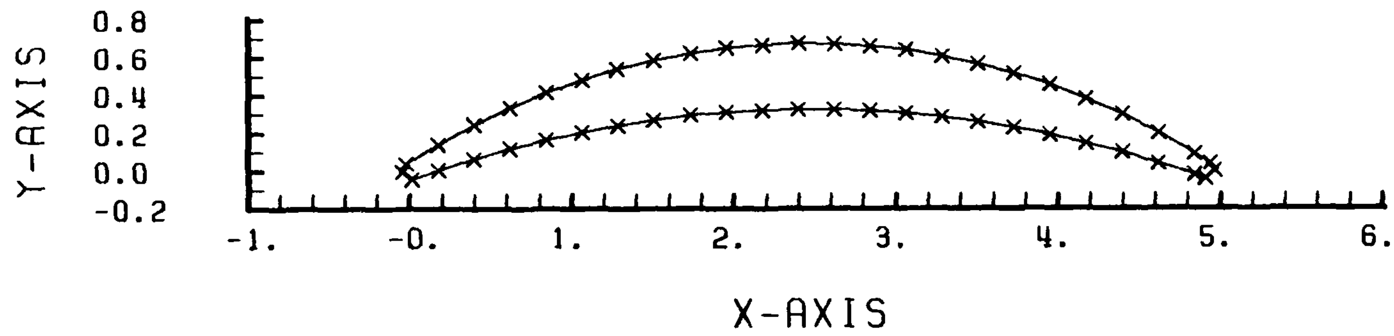


Fig. D.1. Blade Coordinates

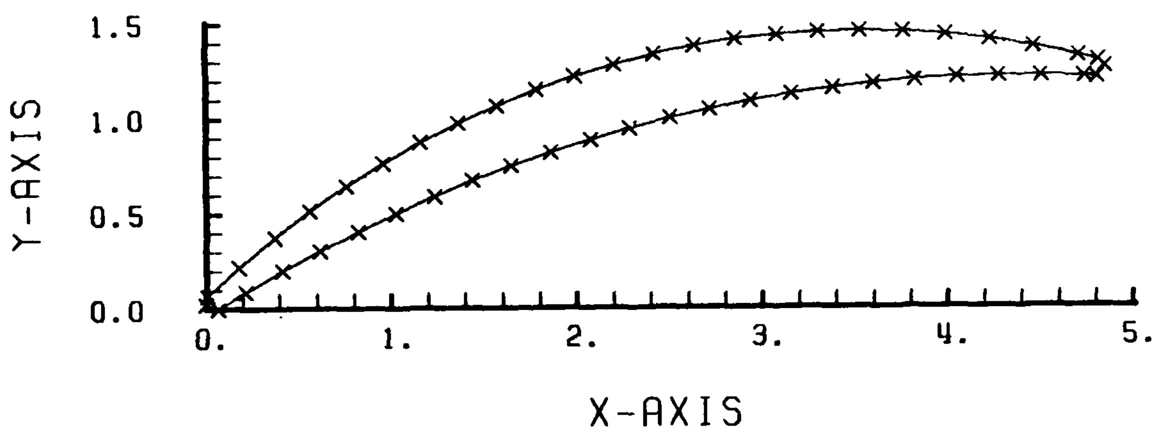


Fig. D.2. Blade Coordinates Translated and Rotated

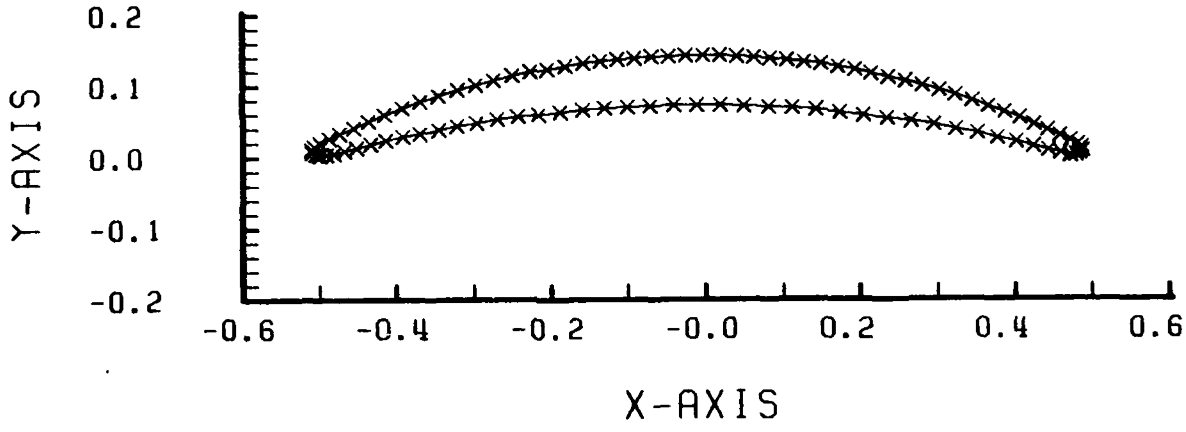


Fig. D.3. Mesh Points on Blade Surface, Horizontal Chord

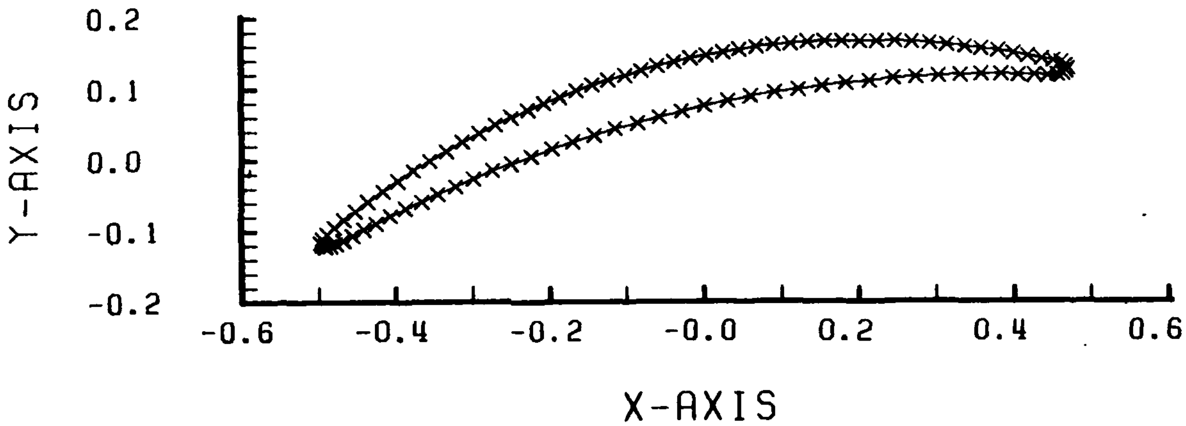


Fig. D.4. Mesh Points on Blade Surface, Chord at Stagger Angle

APPENDIX E

QSONIC PROGRAM LISTING

```
C THIS VERSION WAS COMPILED USING FORTGI. THE SAME CODE COMPILED
C  WITH FORTHX GAVE IDENTICAL RESULTS.
C                                 QSONIC
C
C      COMPUTER PROGRAM FOR CALCULATING THE FULL POTENTIAL QUASI-3D
C
C      TRANSONIC FLOW THROUGH A ROTATING TURBOMACHINERY BLADE ROW.
C
C                  C. A. FARRELL,   J. J. ADAMCZYK
C                  NASA LEWIS RESEARCH CENTER  1981
C        ADAPTED TO THE NAVAL POSTGRADUATE SCHOOL IBM 370/3033
C                                BY
C                 LT. WILLIAM D. MOLLOY JR.  18 DEC 81
C
C                          QSONIC EXECUTIVE
C
C                    READS CASE CONTROL INPUT
C                    FOR EACH GRID REQUESTED, CALLS
C                        GEOMETRY SETUP
C                        STREAMCHANNEL DATA SETUP
C                        FLOW SOLVER
C                    CALLS FINAL OUTPUT GENERATOR
C
      LOGICAL RESTAR,NOFLOW,QUASI3,LEONLY
     1,BUG,BUG2,BUG3,SMOOTH,FINEST,ALLOUT
      INTEGER REMESH,CAMPER
      REAL MINF,LAMDAO,MINF2
      DOUBLE PRECISION CAPK,CAPKP
      COMMON/MESHES/NOZES(10),MS(10),TOLS(10),NOLD(2)
      COMMON/PARAM/RESTAR,NOZIN,BETA1,BETA2,QINF1,CI,CII,GAMMA,MINF,
     1MINF2
     1,DELTA,BETA,TOL,BUG,BUG2,BUG3,IT,GUESS,OVEREL,TIMER,NOWREL,NOTYET
     1,DAMP,SUPREL,S,WAKE,B2OB1,UNDERL,R2OR1,RINF,OMEGA,FLOCO,VAXIAL
     1,ALLOUT,KKMAX
      COMMON/QUASI/R(100,30),B(100,30)
      COMMON/ENTIRE/  X(100,30),Y(100,30),PHI(100,30),DELPSV(100,30)
      COMMON/GEOM2/ZETA(100),ETA(100)
      COMMON/GEOM/NEWNOZ,NEWM,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK,
     1CAPKP,PI,RNK,VB,KN,NED,BB,DOSURF,RFAC,SOLID,MPLS
     1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THETT,THETL
     1,FINEST,LEONLY,CAMPER

      DIMENSION TITLE(20)
      NAMELIST/PARAMS/ NOFLOW,RESTAR,REMESH,MS,NOZES,MESH1,MESHN,
     1LAMDAO,CHORD,S,NED,KN,RLE,RTE,THETL,THETT,CAMPER,STABAC,CHOP,
     1SMOOTH,LEONLY,SLP1,SLP2,SLP3,SLP4,MINF,BETA1,BETA2,GAMMA,
     1TOLS,OVEREL,UNDERL,SUPREL,NOWREL,NOTYET,TEGARD,DAMP,CII,IT,
     1ALLOUT,QUASI3,NSTRM,RINF,WAKE,OMEGA,VAXIAL,FLOCO,MINF2
```

```
      DATA NSTRM,QUASI3,MESH1/1,.FALSE.,1/,REMESH/4/,TEGARD/2./
     1,NOFLOW/.FALSE./

      DATA QLEVEL/'9.1'/

      READ(5,3) TITLE
    3 FORMAT(20A4)
      WRITE(6,33) QLEVEL,TITLE
   33 FORMAT(1H1,' QSONIC VERSION 1.4.',A4,2X,21A4//)


      READ(5,PARAMS)
      WRITE(6,2)
    2 FORMAT('0     CASE CONTROL INPUT ECHO '/)
      WRITE(6,PARAMS)


      BETA1=BETA1*PI/180.
      LAMDA0=LAMDA0*PI/180.
      BETA2=BETA2*PI/180.
      MIN=MS(MESH1)
      IF(RESTAR)MESH1=REMESH
      NOZIN=IABS(NOZES(MESH1))
      IF(QUASI3.AND.B2OB1.EQ.1.)CALL RBIN(NSTRM,LAMDA0)


      DO 1 I=MESH1,MESHN
      MESH=I
      NOLD(2)=NEWM
      NOLD(1)=IEXIT
      NEWNOZ=NOZES(MESH)
      NEWM=MS(MESH)
      TOL=TOLS(MESH)
      REWIND 23
      REWIND 13
      CALL WRAPUP(ZETA,ETA,S)
      IEXIT=2*(NEWNOZ-1)
      CALL WHEEL(LAMDA0,NEWM,IEXIT,NOFLOW)
      IF(NOFLOW)STOP
      IF(QUASI3) CALL FILLRB(NEWM,IEXIT,RINF,LAMDA0)
      IF(RESTAR.OR.MESH.NE.MESH1)CALL FILPHI(NEWNOZ,NEWM,IEXIT)
      IF(SLP1.GE.0.)SLP2=TEGARD
      CALL USONIC
      MYMESH=MESH-MESH1+1
      MYMAX=MESHN-MESH1+1
      WRITE(6,4) MYMESH,MYMAX
    4 FORMAT(1H0,40X,' END OF CALCULATION ON GRID',I2,' OF',I2)
```

```
1     CONTINUE


      CALL KEEPER
      STOP
      END
      BLOCK DATA

C               INITIALIZE ALL DATA IN COMMON

      LOGICAL RESTAR,LEONLY
     1,BUG,BUG2,BUG3,SMOOTH,FINEST,ALLOUT
      INTEGER CAMPER
      REAL MINF,LAMDAO
      DOUBLE PRECISION CAPK,CAPKP
      COMMON/MESHES/NOZES(10),MS(10),TOLS(10),NOLD(2)
      COMMON/PARAM/RESTAR,NOZIN,BETA1,BETA2,QINF1,CI,CII,GAMMA,MINF,EM
     1,DELTA,BETA,TOL,BUG,BUG2,BUG3,IT,GUESS,OVEREL,TIMER,NOWREL,NOTYET
     1,DAMP,SUPREL,S,WAKE,B2OB1,UNDERL,RADRAT,RADLE,OMEGA,FLOCO,VAXIAL
     1,ALLOUT,KKMAX
      COMMON/QUASI/R(100,30),B(100,30)
      COMMON/ENTIRE/   X(100,30),Y(100,30),PHI(100,30),DELPSV(100,30)
      COMMON/GEOM2/ZETA(100),ETA(100)
      COMMON/GEOM/NEWNOZ,NEWM,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK,
     1CAPKP,PI,RNK,VB,KN,NED,BB,DOSURF,RFAC,SOLID,MPLS
     1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THETT,THETL
     1,FINEST,LEONLY,CAMPER

      COMMON/CALVEL/D55(55),PRIORS(14,30),INDEX(100,30),I1DEX(32),
     1FAKEU(100),FAKEV(100),ICOUNT,IDEX,AMY,AMZ

      DATA RADRAT/1./,WAKE,B2OB1/0.,1./,R,B/6000*1./
      DATA RADLE/1./,OVEREL/1.5/,TOL/.001/
     1,PI/3.14159265/,TIMER/1./
     1,OMEGA,FLOCO,VAXIAL/0.,999.,0./
     1,SUPREL/1./,CHOP/.99/,THETT,THETL/0.,0./,CAMPER/6/,EM/10./
      DATA NOTYET/2/,DAMP/1./,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH/.999,
     11.,2.,2*1.,.FALSE./,UNDERL/1./,LEONLY/.FALSE./,FINEST/.FALSE./
      DATA A/1.5/,CII/.2/,CI/1./,GUESS/10./,IT/10/,ALLOUT/.FALSE./
      DATA KKMAX/300/
      DATA DELTA,BETA,QINF1,BUG,RESTAR,GAMMA/3*1.,.FALSE.,.FALSE.,1.4/


      DATA INDEX,I1DEX/203*0,1,98*0,2,1,96*0,2,2,4,1,96*0,1,1,1,2497*0,
     1 2,8,2,8,1,1,8,3,3,3,1,1,1,8,2,8,2,8,1,1,8,3,3,3,1,1,1,8,2,8,2,8/
      END
```

```
      SUBROUTINE WHEEL(LAMDAO,NEWM,NEW2M1,NOFLOW)
C     ROTATE X,Y COORDINATES THRU AN ANGLE LAMDAO(RADIANS)

      LOGICAL NOFLOW
      COMMON/ENTIRE/  X(100,30),Y(100,30),PHI(100,30),DELPSV(100,30)
      COMMON/GEOM2/ZETA(100),ETA(100)
      REAL LAMDAO
      COMPLEX H,AI
      DATA AI/(0.,1.)/
      NEW2M=NEW2M1-1
      MM3=NEWM-3
      NNOZ=(NEW2M+1)/2
      NEW2M2=NEW2M1+1
      DO 1 I=1,NEW2M2
      DO 1 J=1,NEWM
      H=CMPLX(X(I,J),Y(I,J))*CEXP(AI*LAMDAO)
      X(I,J)=REAL(H)
1     Y(I,J)=AIMAG(H)
      IF(.NOT.NOFLOW) RETURN
      WRITE(6,10)
10    FORMAT(1H1,30X,' GRID OUTPUT POINTS ON BLADE SURFACE, CHORD ',
     1'AT STAGGER ANGLE'//42X,'  I        X                Y'/)
      DO 2 I=1,NEW2M2
      I1M=I-1
2     WRITE(6,14) I1M,X(I,NEWM),Y(I,NEWM)
14    FORMAT(41X,I4,2E13.5)
      RETURN
      END
      SUBROUTINE FILPHI(NEWNOZ,NEWM,NEW2M1)
C        BASED ON PHI ARRAY FROM PREVIOUS GRID, INTERPOLATES VALUES
C        TO FILL THE INITIAL PHI ARRAY FOR NEXT, FINER GRID.

      INTEGER OLDNOZ,OLDM
      LOGICAL RESTAR
      COMMON/ENTIRE/  X(100,30),Y(100,30),PHI(100,30),TOPHI(100,30)
      COMMON/MESHES/NOZES(10),MS(10),TOLS(10),NOLD(2)
      COMMON/GEOM2/ ZNEW(100),ENEW(100)
      COMMON/CALVEL/FAKEFI(4,6),FAKFI(4,6),RHOCON,QLIM,EXPO,ROTATN,
     1ROTROT,IM1,IEXIT,PRIORS(14,30),INDEX(100,30),I1DEX(32),FAKEU(75)
     1,FAKEV(75),ICOUNT,IDEX
      COMMON/PARAM/ RESTAR
      DIMENSION OLDE(100),OLDZ(100)
```

```
                                                                   QS001930
      OLDNOZ=NOLD(1)                                               QS001940
      OLDM=NOLD(2)                                                 QS001950
      NZOLD=OLDNOZ/2+1                                             QS001960
      INDEX(OLDNOZ-2,3)=0                                          QS001970
      INDEX(OLDNOZ-2,4)=0                                          QS001980
      INDEX(OLDNOZ-2,5)=0                                          QS001990
      INDEX(OLDNOZ-2,6)=0                                          QS002000
      INDEX(OLDNOZ-1,5)=0                                          QS002010
      INDEX(OLDNOZ-1,6)=0                                          QS002020
      INDEX(OLDNOZ,5)=0                                            QS002030
      INDEX(OLDNOZ,6)=0                                            QS002040
      INDEX(NZOLD-2,3)=0                                           QS002050
      INDEX(NZOLD-2,4)=0                                           QS002060
      INDEX(NZOLD-2,5)=0                                           QS002070
      INDEX(NZOLD-2,6)=0                                           QS002080
      INDEX(NZOLD-1,5)=0                                           QS002090
      INDEX(NZOLD-1,6)=0                                           QS002100
      INDEX(NZOLD,5)=0                                             QS002110
      INDEX(NZOLD,6)=0                                             QS002120
      INDEX(NZOLD+1,4)=0                                           QS002130
      INDEX(NZOLD+1,5)=0                                           QS002140
      INDEX(NZOLD+1,6)=0                                           QS002150
      INDEX(NZOLD+2,3)=0                                           QS002160
      INDEX(NZOLD+2,4)=0                                           QS002170
      INDEX(NZOLD+2,5)=0                                           QS002180
      DO  13 I1=1,NEW2M1                                           QS002190
      DO 13  J1=1,NEWM                                             QS002200
13    TOPHI(I1,J1)=0.                                              QS002210
      REWIND 18                                                    QS002220
      REWIND 8                                                     QS002230
                                                                   QS002240
                                                                   QS002250
      IF(RESTAR)GOTO10                                             QS002260
      READ(18) OLDE,OLDZ                                           QS002270
      DO 1  I1=1,NEW2M1                                            QS002280
      DO 2  I2=1,OLDNOZ                                            QS002290
      IF(OLDE(I2).GT.ENEW(I1)) GOTO3                               QS002300
2     CONTINUE                                                     QS002310
      I2=OLDNOZ                                                    QS002320
3     DO 1  I3=1,NEWM                                              QS002330
      DO 5  I4=1,OLDM                                              QS002340
      IF(OLDZ(I4).GT.ZNEW(I3))GOTO6                                QS002350
5     CONTINUE                                                     QS002360
      I4=OLDM                                                      QS002370
6     FACE=(ENEW(I1)-OLDE(I2-1))/(OLDE(I2)-OLDE(I2-1))             QS002380
      TERP3=                                                       QS002390
     1PHI(I2-1,I4)+(PHI(I2,I4)-PHI(I2-1,I4))*FACE                  QS002400
```

```
      TERP1=
     1PHI(I2-1,I4-1)+(PHI(I2,I4-1)-PHI(I2-1,I4-1))*FACE
      BOP=(TERP3-TERP1)/(OLDZ(I4)-OLDZ(I4-1))*(ZNEW(I3)-
     1OLDZ(I4-1))+TERP1
1     TOPHI(I1,I3)=BOP


      DO 7 I1=1,NEW2M1
      DO 7 I3=1,NEWM
7     PHI(I1,I3)=TOPHI(I1,I3)


22    FORMAT(4I6,4E13.5)
4     FORMAT(1H ,8E13.5)
      WRITE(8,11) PHI
      DO  1313 I1=1,NEW2M1
      DO 1313 J1=1,NEWM
1313  TOPHI(I1,J1)=0.
      RETURN

C       IF RESTART, READ PHI FROM THE RESTART FILE
10    READ(9,11) PHI
11    FORMAT(6E13.5)
      RESTAR=.FALSE.
      REWIND 9
12    RETURN
      END
      SUBROUTINE RBIN(NSTRM,LAMDAO)

C  FOR QUASI3D CASES, READS UNIT 2 WITH DISTRIBUTIONS OF SPANWISE RADIUS
C    AND CHANNEL THICKNESS.  FORMAT IS THE SAME AS PROGRAM TSONIC.

      REAL LAMDAO
      COMMON/MERIDL/ ZMSP(50,2),THSP(50,2),RMSP(50),
     1BESP(50),RM(50),NRSP,NSP(2),RTE(2),RLE(2),STGRF,CHO,CNBL,DM
      DIMENSION DUM1(50),R(2)
      REWIND 2
      ISTRM=IABS(NSTRM)
      DO 1 I=1,ISTRM
      READ(2,3)
      READ(2,3)
      READ(2,6) A1,A2,CHO,STGRF
      READ(2,3)
3     FORMAT(1H )
      READ(2,4) NBL,NRSP,I1,I2,I3
      CNBL=FLOAT(NBL)
4     FORMAT(30X,I5,4I5)
      DO 10 J=1,2
```

```
      READ(2,5) RI,RO,SPLNO1
      ISPL1=SPLNO1
C     WRITE(6,4) I,NRSP,J,ISPL1
      NSP(J)=ISPL1
      RTE(J)=RO
      RLE(J)=RI
5     FORMAT(2F10.5,20X,F10.5)
6     FORMAT(8F10.6)
      READ(2,6) (ZMSP(IA,J),IA=1,ISPL1)
10    READ(2,6) (THSP(IA,J),IA=1,ISPL1)

C    READ X VALUES

      READ(2,6) (RM(IA),IA=1,NRSP)

C    READ CORRESPONDING SPANWISE RADIUS VALUES.

      READ(2,6) (RMSP(IA),IA=1,NRSP)

C    READ STREAMCHANNEL THICKNESS VALUES.

      READ(2,6) (BESP(IA),IA=1,NRSP)

      IF(I1.EQ.1)READ(2,6) (DUM1(IA),IA=1,NRSP)
      IF(I2.EQ.1)READ(2,6) (DUM1(IA),IA=1,NRSP)
1     READ(2,4) I4
      IF(NSTRM.LT.0) RETURN

C     TRANSLATE ORIGIN OF X VALUES TO MIDCHORD AND NORMALIZE RADIUS AND
C     THICKNESS TO THE AERODYNAMIC CHORD.

      DO 7 I=1,NRSP
      RM(I)=(RM(I)-CHO/2.)/CHO*COS(LAMDAO)
      RMSP(I)=RMSP(I)/CHO*COS(LAMDAO)
7     BESP(I)=BESP(I)/CHO*COS(LAMDAO)


      RETURN
      END
      SUBROUTINE FILLRB(NEWM,NEW2M1,RADLE,STAG)

C INTERPOLATES ON THE INPUT (OR ASSUMED LINEAR) DISTRIBUTIONS OF RADIUS
C AND THICKNESS TO FILL ARRAYS OF SPANWISE POSITION AND STREAMCHANNEL
C THICKNESS AT ALL GRID POINTS.

C            IF RADLE=1.,  RADIUS IS ASSUMED CONSTANT
```

```
      COMMON/QUASI/R(100,30),B(100,30)
      COMMON/MERIDL/ ZMSP(50,2),THSP(50,2),RMSP(50),
     1BESP(50),RM(50),NRSP,NSP(2),RTE(2),RLE(2),STGRF,CHO,CNBL,DUMM
      COMMON/ENTIRE/  X(100,30),Y(100,30),PHI(100,30),DELPSV(100,30)
      COMMON/PARAM/ RESTAR,NOZIN,BRM(17),USELER,BR(6),B2OB1,UNDERL,R2OR1
      NEW2M2=NEW2M1+1
      N=NRSP
      CO=COS(STAG)/2.

C            NOZIN=NEW2M1/2 ON THE FIRST PASS ONLY

      IF(NOZIN.EQ.NEW2M1/2)FIXR=RADLE
      IF(NOZIN.EQ.NEW2M1/2)B21SAV=B2OB1
      IF(RADLE.NE.1..AND.R2OR1.EQ.1.)R2OR1=RMSP(NRSP)/RMSP(1)
      IF(B2OB1.EQ.1.)B2OB1=BESP(NRSP)/BESP(1)


      IF(B21SAV.NE.1..OR.NOZIN.NE.NEW2M1/2.OR.USELER.EQ.0..OR.RADLE.EQ.1
     *.)GOTO15

C          FIND L.E. OF BLADE (RM+CO=0.)

      DO 13 I5=1,NRSP
      IF(RM(I5)+CO.GT.0.)GOTO14
13    CONTINUE
      I5=NRSP

C     CALCULATE SPANWISE POSITION AT L.E. (FIXR)  FROM THE GIVEN RADIUS
C     DISTRIBUTION.
14     FIXR=
     1RMSP(I5-1)-(RMSP(I5)-RMSP(I5-1))*(CO+RM(I5-1))/(RM(I5)-RM(I5-1))


15    DO 1 I1=1,NEW2M2
      DO 1 I3=1,NEWM
      IF(B21SAV.NE.1.)GOTO12
      DO 2 I2=1,NRSP
      IF(RM(I2).GT.X(I1,I3))GOTO3
2     CONTINUE
      I2=NRSP
3     IF(I2.EQ.1)I2=2
      FACE=(X(I1,I3)-RM(I2-1))/(RM(I2)-RM(I2-1))
      TERPR=RMSP(I2-1)+(RMSP(I2)-RMSP(I2-1))*FACE
C     IF(RM(NRSP).LT.X(I1,I3))TERPR=RMSP(NRSP)
C     IF(RM(1).GT.X(I1,I3))TERPR=RMSP(1)
      TERPB=BESP(I2-1)+(BESP(I2)-BESP(I2-1))*FACE
      IF(RADLE.NE.1.)R(I1,I3)=TERPR
```

```
      B(I1,I3)=TERPB/BESP(1)
      GOTO10

C  IF (B21SAV.NE.1), A LINEAR DISTRIBUTION OF THICKNESS IS ASSUMED OVER
C   3 CHORD LENGTHS.

12    B(I1,I3)=1.+(B2OB1-1.)*(X(I1,I3)+1.5)/3.
      IF(RADLE.NE.1.)FIXR=(1.+(R2OR1-1.)*(1.5-CO)/3.)*RADLE
      IF(X(I1,I3).LT.-1.5)B(I1,I3)=1.
      IF(X(I1,I3).GT.1.5)B(I1,I3)=B2OB1
      IF(RADLE.NE.1.)R(I1,I3)=(1.+(R2OR1-1.)*(X(I1,I3)+1.5)/3.)*RADLE
      IF(X(I1,I3).LT.-1.5)R(I1,I3)=RADLE
      IF(X(I1,I3).GT.1.5)R(I1,I3)=R2OR1*RADLE



C  SCALE THE TANGENTIAL COORDINATE Y (R*THETA), TO THE LOCAL RADIUS.

C10   Y(I1,I3)=Y(I1,I3)*R(I1,I3)/FIXR
10    CONTINUE
1     CONTINUE
      USELER=FIXR
      RETURN
4     FORMAT(2I5,6E10.4)
      END

      SUBROUTINE USONIC



C                           FULL POTENTIAL FLOW SOLVER
C                             FOR CASCADE GEOMETRIES

C       ETA = COMPUTATIONAL COORDINATE THAT  IS CONSTANT ON LINES
C             RADIATING FROM BLADE .
C       ZETA= COMPUTATIONAL COORDINATE THAT IS CONSTANT ON LINES
C             SURROUNDING THE BLADE.
C             (THESE ARE REVERSED FROM INTERPOLATION GRID GENERATOR)


C     DO (INPUT SPECIFICATION)
C     DO (INITIALIZATION OF FIELD VARIABLES)
C     RELAXD= .FALSE.
C     DO UNTIL (RELAXD)
C        KK=KK+1
C        KKK=KKK+1
C        IF(KK.GT.NOWREL) OVEREL=RELSAV
C        IF(KK.GT.NOTYET) SUPREL=SUPSAV
```

```
C         DO UNTIL (EXIT)
C            I=I+1
C            IM1= I-1
C            IF(IM1.EQ.0) IM1=2
C            DO (AXIAL LOCATION INDICATORS)
C            DO UNTIL (TOP)
C               J=J+1
C               JM1=J-1
C               IF(JM1.EQ.0) JM1=2
C               DO (VERTICAL LOCATION INDICATORS)
C               DO (BOUNDARY CONDITION SWITCHES)
C               DO (GEOMETRY ADVANCE)
C               DO (FIELD GRADIENTS)
C               DO (VELOCITIES AND DENSITIES)
C               DO (SUPERSONIC CORRECTION)
C               DO ( T'S,Q'S AND RESIDUALS)
C               DO ( DELPHI COEFFICIENTS AND TRANSPOSED TERMS)
C            END
C            DO (FORWARD ELIMINATION)
C            IF(I.NE.1)
C           1DO (BACK SUBSTITUTION)
C            J=0
C         END
C         DO (UPDATE OF FIELD VARIABLES)
C         DO (CONVERGENCE TESTS)
C         I=0
C      END
C      DO (SURFACE FLOW CALCULATION FOR FINISHED GRID)
C      RETURN
C      ENTRY KEEPER
C      DO (OUTPUT OF PHI AND CALCULATION OF FINAL VELOCITY FIELD)
C      RETURN


       LOGICAL INLET,BLADE,EXIT,BOTTOM,TOP,
      1  RELAXD,SETTLE,BUG,   BUG3,BUG2
       LOGICAL DOSURF,SMOOTH,FINEST,RESTAR,ALLOUT
       REAL LAB,LAD,MINF,LENGTH,JACOB,LAMDAO
       REAL MACHA,MACHB,MINF2
       REAL MACH(8)
       INTEGER FIRSTJ
       DOUBLE PRECISION CAPK,CAPKP
       COMMON/PARAM/RESTAR,NOZIN,BETA1,BETA2,QINF1,CI,CII,GAMMA,MINF,EM
      1,DELTA,BETA,TOL,BUG,BUG2,BUG3,IT,GUESS,OVEREL,FIXR,NOWREL,NOTYET
      1,DAMP,SUPREL,S,WAKE,B2OB1,UNDERL,RADRAT,RADLE,OMEGA,FLOCO,VAXIAL,
      1ALLOUT,KKMAX
       COMMON/QUASI/ RADIUS(100,30),HEIGHT(100,30)
       COMMON/ENTIRE/  X(100,30),Y(100,30),PHI(100,30),DELPSV(100,30)
```

```
      COMMON/GEOM2/ZETA(100),ETA(100)
      COMMON/CROSS/ ZMACH(100,30)
      COMMON/GEOM/NOZ,M,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK,
     1CAPKP,PI
     1,RNK,VB,KN,NED,BB,DOSURF,RFAC,SOLID,MPLS
     1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THETT,THETL
     1,FINEST
      COMMON/CALVEL/FAKEFI(4,6),FAKFI(4,6),RHOCON,QLIM,EXPO,ROTATN,
     1ROTROT,IM1,IEXIT,PRIORS(14,30),INDEX(100,30),I1DEX(32),FAKEU(100)
     1,FAKEV(100),ICOUNT,IDEX,AMY,AMZ
      COMMON/VELOUT/D(100),E(100),F(100),DELPHI(100),TENOLF(100,30),
     1DUMBX(100,30),DUMBY(100,30),BOPR,FADBOP,XOCX,KK
      DIMENSION RHOSS(8), FI1(8),FI2(3),FI3(3),FI9(8),SPAN(8),
     1EMACH(100,30)
     2,COSX(8),COSY(8),LENGTH(8),DEDX(8),DZDX(8),DEDY(8),
     3DZDY(8),T(8),DPHIDZ(8),DPHIDE(8),U(8),V(8),UN(8),RHO(8),
     4RHOUL(8),ONOFF(8),DELPM1(3)
     5,HITE(8)
      NAMELIST/ABSLUT/MINF,MINF2,QINF2,ABETA1,ABETA2,AQINF1,AQINF2,AMINF
     1,AEM,ROTATN
      NAMELIST/PARAMS/     IEXIT,NOZ,M,BETA1,BETA2,LAMDAO
     1,S,CHORD,RLE,RTE,NED,KN
     1,STABAC,SLP1,SLP2,SLP3,SLP4,CHOP,THETT,THETL
     1,QINF1,CI,CII,GAMMA,MINF,EM,DELTA,BETA,WAKE,B2OB1,RADRAT
     1,OMEGA,FLOCO,VAXIAL
     1,TOL,BUG,IT,GUESS,OVEREL,   SUPREL,NOWREL,NOTYET,DAMP,
     1 SMOOTH,FINEST,BUG,BUG2,BUG3
      NAMELIST/YTESD/ KK,AIJ,AIJM1,AIJP1, TRANSP,I,J,CIRCO
      NAMELIST/CONST/CIRCLN,MINF2,B2OB1,R2OR1,RINF,RATLE
      DATA AQINF1/1./,AVGFAC,TEST/1.,1./

      COMMON/HIBALL/ SUPERG(46,30,100)

      XDIF(A,B,C,D)=.5*(A+B-C-D)

C     DO (INPUT SPECIFICATION)

C     PROCEDURF (INPUT SPECIFICATION)

      IF(NOZ-1.NE.NOZIN) GOTO 201
C     IF(NOZ-1.EQ.NOZIN) THEN
```

```
      RELSAV=OVEREL
      UNDSAV=UNDERL
      SUPSAV=SUPREL
      SHIFTM=ABS(CI)
      IF(RADRAT.EQ.1.)RADLE=1.
      IF(RADRAT.EQ.1.)FIXR=1.
      IF(SHIFTM.EQ.0.)SHIFTM=1.
      IF(FLOCO.EQ.999..AND.OMEGA.NE.0.)
     1FLOCO=VAXIAL/OMEGA/RADLE
      IF(FIXR.LT.1.E-06)FIXR=1.
      ROTATN=COS(BETA1+LAMDA0)/FLOCO
      IF(OMEGA.EQ.0..AND.FLOCO.EQ.999.)ROTATN=0.
      ROTROT=ROTATN**2
      BEETA1=(BETA1+LAMDA0)
      BEETA2=(BETA2+LAMDA0)


C     DO (EXIT MACH NUMBER ITERATION)


C     PROCEDURE (EXIT MACH NUMBER ITERATION)

      FAB=(GAMMA-1.)/2.
      ROTFAC=ROTROT*(RADRAT**2-1.)
      IF(EM.NE.10.) GOTO205
C     IF(EM.EQ.10.) THEN
      P1OP2=1./(RADRAT-WAKE/S)
      SETTLE=.FALSE.
      IF(MINF.GT.0.)EM=MINF*.8
      IF(MINF.LT.0.)EM=1.0+ABS(MINF)
      MINF=ABS(MINF)
      FAC1=1./(1.+FAB*MINF*MINF*(1.+ROTFAC))
      FAC1=FAC1**((GAMMA+1.)/4./FAB)*COS(BEETA1)/COS(BEETA2)/B2OB1
     1*P1OP2*MINF
208   FAC2=1./(1.+FAB*EM*EM)
      FAC3=FAC2**((GAMMA+1.)/4./FAB)
      EFF=EM*FAC3-FAC1
      DFDM=FAC3*(1.-FAC2*EM*EM*(GAMMA+1.)/2.)
      DM=-EFF/DFDM
      SETTLE=ABS(DM)/EM.LT..001
      EM=EM+DM
      IF(.NOT.SETTLE)GOTO208
C     END
205   MINF2=EM
      MINF=ABS(MINF)
      QINF2=SQRT((1.+ROTFAC+1./FAB/MINF/MINF)/(1.+1./FAB/EM/EM))
      AQINF2=QINF2
      ABETA2=BEETA2*180./PI
```

```
      ABETA1=BEETA1*180./PI
      IF(ROTATN.EQ.0.) GOTO210
C     IF(ROTATN.NE.0.) THEN
      SINBET=SIN(BEETA1)+ROTATN
      COSBET =COS(BEETA1)
      ABETA1=ATAN2(SINBET,COSBET)*180./PI
      SINBET=QINF2*SIN(BEETA2)+ROTATN*RADRAT
      COSBET=QINF2*COS(BEETA2)
      ABETA2=ATAN2(SINBET,COSBET)*180./PI
      AQINF1=COS(BEETA1)/COS(PI/180.*ABETA1)
      AQINF2=QINF2*COS(BEETA2)/COS(PI/180.*ABETA2)
      AMINF=MINF*COS(BEETA1)/COS(PI/180.*ABETA1)
      AEM=EM*COS(BEETA2)/COS(ABETA2*PI/180.)
      WRITE(6,779)
779   FORMAT('1ROTATION EFFECTS INCLUDED,ABSOLUTE AND RELATIVE VELOCITY'
     1,' TRIANGLES HAVE - ')
      WRITE(6,ABSLUT)
      BETA1=ABETA1*PI/180.-LAMDAO
      BETA2=ABETA2*PI/180.-LAMDAO
C     END
210   P1OP2=1./RADRAT
      CIRCO=RADLE/FIXR*S*(AQINF1*SIN(LAMDAO+BETA1)-AQINF2*SIN(LAMDAO+
     1BETA2)/P1OP2)
      CIRCOO=CIRCO
      IF(CIRCO.EQ.0.)CIRCOO=1.
      CIRCLN=CIRCO
      R2OR1=RADRAT
      RINF=RADLE
      RATLE=FIXR
      WRITE(6,780)
780   FORMAT('1CALCULATED FLOW PARAMETERS')
      WRITE(6,CONST)
C      END
C     END


201   IF(NOZ-1.NE.NOZIN)SUPSAV=SUPSAV-CII
      IF(SUPSAV.LT..5)SUPSAV=.5
      OVEREL=1.
      UNDERL=1.
      SUPREL=1.
      NZZ=2*NOZ-2
      KKK=0
      IEXIT=NZZ
      I67=M-14+1
      I67=MAXO(I67,1)
2     FORMAT(6E13.5)
```

```
      KK=0
      M3=M-3
      ITIEX=IEXIT-1
      WRITE(6,777) M3,ITIEX,SUPSAV
777   FORMAT(1H1,12X,
     1'RELAXATION BEGINS ON GRID OF ',I3,' SURFACE CONTOURS ',
     1'INTERSECTED BY ',I3,' RADIATING LINES'/1H0,30X,'SUPERSONIC ',
     1'RELAXATION ',
     1'PARAMETER, SUPREL= ',E13.5)
      WRITE(6,778)
778   FORMAT(1H0,3X,'ITERATION    AVERAGE       RESIDUAL     ',
     1' MAX MACH     ',
     1' CALCULATED   RELATIVE CIRC     SUBSONIC    SUPERSONIC    '/6X,
     1'COUNT     PHI CORRECTN     AT T. E.      ON BLADE      CIRCULATION',
     1'       ERROR     ',
     1'    RELAX. FAC.  RELAX. FAC.'/)
C       IF(WAKE.EQ.0.) GOTO212
CC      IF(WAKE.NE.0.) THEN
C       XMAX=0.
C       DO 101 I1=1,IEXIT
C       IF(X(I1,M).GT.XMAX)XMAX=X(I1,M)
C       IF(X(I1,M).GT.XMAX)ITE=I1
C101    CONTINUE
C       X44=.95*X(ITE,M)
C       X12=(X(2,4)-X44)*SLP3
C       MO2=M/2
C       DO 109 I105=1,IEXIT
C       IF(ABS(X(I105,MO2)-X12-X44).LE.1.E-02) GOTO1091
C109    CONTINUE
C1091   CONTINUE
C       S105=RADLE/FIXR*S*RADIUS(I105,MO2)/RADLE
C       BDELT=WAKE/S105*SLP4
C       DO 119 I=1,IEXIT
C       DO 118 J=1,M
C       IF(X(I,J).GT.X44)TERPIT=AMIN1((X(I,J)-X44)/X12,1.)
C       IF(X(I,J).GT.X44)HEIGHT(I,J)=HEIGHT(I,J)*(1.-TERPIT*BDELT)
C118    CONTINUE
C119    CONTINUE
C     END
C       END



C       DO (INITIALIZATION OF FIELD VARIABLES)


C       PROCEDURE (INITIALIZATION OF FIELD VARIABLES)

```

```
212   COBB=COS(LAMDAO+BETA1)*AQINF1*RADLE                                QS006730
      COSB=COS(LAMDAO+BETA1)*AQINF1                                      QS006740
      SIBB = SIN(LAMDAO+BETA1)*AQINF1*RADLE                              QS006750
      SINB=SIN(LAMDAO+BETA1)*AQINF1                                      QS006760
      IF(RESTAR.OR.NOZ-1.NE.NOZIN)GOTO219                                QS006770
      IEXP1=IEXIT+1                                                      QS006780
      BBLR= BEETA1+BEETA2+LAMDAO+ROTATN                                  QS006790
      DO 135 I=1,IEXP1                                                   QS006800
      DO 134  J=1,M                                                      QS006810
      PHI(I,J)=COSB*X(I,J)+SINB*Y(I,J)                                   QS006820
      IF(ROTATN.NE.O.)PHI(I,J)=PHI(I,J)*RADLE                            QS006830
      RDDDD=RADLE/HEIGHT(I,J)/RADIUS(I,J)                                QS006840
      IF(BBLR.EQ.0.)PHI(I,J)=PHI(I,J)*RDDDD                              QS006850
134   CONTINUE                                                           QS006860
135   CONTINUE                                                           QS006870
219   NOZM1=NOZ-1                                                        QS006880
      NOZP1=NOZ+1                                                        QS006890
      DO 141 I=NOZM1,NOZP1                                               QS006900
      DO 140 J=3,5                                                       QS006910
      PHI(I,J)=COSB*X(I,J)+SINB*Y(I,J)                                   QS006920
      IF(ROTATN.NE.O.)PHI(I,J)=PHI(I,J)*RADLE                            QS006930
      RDDDD=RADLE/HEIGHT(I,J)/RADIUS(I,J)                                QS006940
      IF(BBLR.EQ.0.)PHI(I,J)=PHI(I,J)*RDDDD                              QS006950
140   CONTINUE                                                           QS006960
141   CONTINUE                                                           QS006970
      DO 153  J=1,6                                                      QS006980
      DO 152 I1=1,4                                                      QS006990
      I10=IEXIT+I1-3                                                     QS007000
      IF(J.LE.5.AND.J.GT.2.AND.                                          QS007010
     1(NOZ-1.NE.NOZIN.OR.RESTAR)                                         QS007020
     1.AND.I1.NE.4)PHI(I1,J)=(COSB*X(I1,J)+SINB*Y(I1,J))                 QS007030
     1                                                                   QS007040
      IF(ROTATN.NE.O.)PHI(I1,J)=PHI(I1,J)*RADLE                          QS007050
      RDDDD=RADLE/HEIGHT(I1,J)/RADIUS(I1,J)                              QS007060
      IF(BBLR.EQ.0.)PHI(I1,J)=PHI(I1,J)*RDDDD                            QS007070
      IF(J.LE.5.AND.J.GT.2.AND.                                          QS007080
     1(NOZ-1.NE.NOZIN.OR.RESTAR)                                         QS007090
     1.AND.I10.GT.IEXIT-2)PHI(I10,J)=(COSB*X(I10,J)+SINB*Y(I10,J))       QS007100
     1                                                                   QS007110
      IF(ROTATN.NE.O.)PHI(I10,J)=PHI(I10,J)*RADLE                        QS007120
      RDDDD=RADLE/HEIGHT(I10,J)/RADIUS(I10,J)                            QS007130
      IF(BBLR.EQ.0.)PHI(I10,J)=PHI(I10,J)*RDDDD                          QS007140
152   CONTINUE                                                           QS007150
153   CONTINUE                                                           QS007160
      DO 161  J=1,M                                                      QS007170
      PHI(1,J)=                                                          QS007180
     1 PHI(IEXIT-1,J)-RADLE/FIXR*S*(AQINF1*SIN(LAMDAO+BETA1)-AQINF2*SIN( QS007190
     1LAMDAO+BETA2)/PIOP2)                                               QS007200
```

```
161    PHI(IEXIT+1,J)=PHI(3,J)+                                          QS007210
     1    RADLE/FIXR*S*(AQINF1*SIN(LAMDA0+BETA1)-AQINF2*SIN(LAMDA0+      QS007220
     1BETA2)/PIOP2)                                                      QS007230
       I=0                                                               QS007240
       J=0                                                               QS007250
       RHOCON=(GAMMA-1.)/2.*MINF*MINF                                    QS007260
       EXPO= 1./(GAMMA-1.)                                               QS007270
       QLIM=1.+1./RHOCON                                                 QS007280
       BMZ=1./MINF/MINF                                                  QS007290
       BMY=(GAMMA-2.)/2.                                                 QS007300
       AMZ=.5/BMZ                                                        QS007310
       BMX=BMY*AMZ/2.                                                    QS007320
       AMY=-BMX/BMZ                                                      QS007330
       BIGK=RADLE/FIXR*COS(BEETA1)                                       QS007340
       UOOUT=AQINF2*COS(ABETA2*PI/180.)                                  QS007350
       DPHTHO=AQINF2*SIN(ABETA2*PI/180.)*RADRAT*RADLE/FIXR               QS007360
       UOIN=AQINF1*COS(ABETA1*PI/180.)                                   QS007370
       DPHTHI=AQINF1*SIN(ABETA1*PI/180.)*RADLE/FIXR                      QS007380
       INDEX(NOZ-2,3)=3                                                  QS007390
       INDEX(NOZ-2,4)=3                                                  QS007400
       INDEX(NOZ-2,5)=3                                                  QS007410
       INDEX(NOZ-2,6)=1                                                  QS007420
       INDEX(NOZ-1,5)=2                                                  QS007430
       INDEX(NOZ-1,6)=1                                                  QS007440
       INDEX(NOZ,5)=2                                                    QS007450
       INDEX(NOZ,6)=1                                                    QS007460
       INDEX(NOZ+1,4)=2                                                  QS007470
       INDEX(NOZ+1,5)=4                                                  QS007480
       INDEX(NOZ+1,6)=1                                                  QS007490
       INDEX(NOZ+2,3)=1                                                  QS007500
       INDEX(NOZ+2,4)=1                                                  QS007510
       INDEX(NOZ+2,5)=1                                                  QS007520
       INDEX(IEXIT-2,3)=3                                                QS007530
       INDEX(IEXIT-2,4)=3                                                QS007540
       INDEX(IEXIT-2,5)=3                                                QS007550
       INDEX(IEXIT-2,6)=1                                                QS007560
       INDEX(IEXIT-1,5)=2                                                QS007570
       INDEX(IEXIT-1,6)=1                                                QS007580
       INDEX(IEXIT,5)=2                                                  QS007590
       INDEX(IEXIT,6)=1                                                  QS007600
                                                                         QS007610
                                                                         QS007620
       IF(B2OB1*RADLE.EQ.1.)GOTO 2691                                    QS007630
C   FOR QUASI - 3D CASES, ADJUST CHANNEL AREA AT FARFIELD ELEMENTS WHERE QS007640
C   IT DROPS BELOW SONIC THROAT AREA BASED ON THE INPUT DISTRIBUTIONS.   QS007650
                                                                         QS007660
       DO 162 I1=1,IEXP1                                                 QS007670
       DO 162 J1=1,M                                                     QS007680
```

```
      IF(INDEX(I1,J1).EQ.0)GOTO162
      ROTFAC=ROTROT*((RADIUS(I1,J1)/RADLE)**2-1.)+1.
      IF(I1.LE.NOZ-3.OR.I1.GE.NOZ+3)GOTO1622
      ASTAR=MINF*COS(BEETA1)/(ROTFAC*MINF*MINF*FAB+1.)**((GAMMA+1.)/4.
     1/FAB)*RADLE*((1.+MINF*MINF*SIN(BEETA1)**2)*FAB+1.)**((GAMMA+1.)/4.
     11/FAB)
      WMACH=MINF
      GOTO1623
 1622 ASTAR=EM*COS(BEETA2)/(ROTFAC*EM*EM*FAB+1.)**((GAMMA+1.)/4.
     1/FAB)*RADLE*((1.+EM*EM*SIN(BEETA2)**2)*FAB+1.)**((GAMMA+1.)/4.
     11/FAB)*RADRAT*B2OB1
      WMACH=EM
 1623 IF(ASTAR.LE.RADIUS(I1,J1)*HEIGHT(I1,J1))GOTO162
      WRITE(6,1666) I1,J1,X(I1,J1),WMACH,ASTAR
 1666 FORMAT('OQUASI-3D CHANNEL CROSSECTION AT BOUNDARY ELEMENT I = ',I3
     1,' J = ',I3,' X = ',E13.5,' FALLS BELOW SONIC THROAT AREA'/
     1' BASED ON MACH= ',F7.4,' AT FAR FIELD STATION,'
     1,' PROCEEDING WITH AREA = 1.01 * THROAT = 1.01 * ',E12.5)

C     RRR=ASTAR/HEIGHT(I1,J1)
C     IF(RADLE.NE.1..AND.B2OB1.NE.1.)RADIUS(I1,J1)=(RRR+RADIUS(I1,J1))/2
C     IF(RADLE.NE.1..AN  .2OB1.NE.1.)HEIGHT(I1,J1)=ASTAR*1.01/RADIUS(I1,
C     IF(B2OB1.NE.1..AN   ADLE.EQ.1.)HEIGHT(I1,J1)=1. 1*ASTAR
C     IF(B2OB1.EQ.1..AND.RADLE.NE.1.)RADIUS(I1,J1)=A. AR*1.01
      HEIGHT(I1,J1)=ASTAR*1.01/RADIUS(I1,J1)
  162 CONTINUE
C     END
 2691 RELAXD= .FALSE.
  269 KK=KK+1
      IDEX=1
      ICOUNT=0
      KKK=KKK+1
      IF(KK.GT.NOWREL) OVEREL=RELSAV
      IF(KK.GT.NOTYET) SUPREL=SUPSAV
      IF(NOZ-1.NE.NOZIN)
     1UNDERL=UNDSAV+FLOAT(KKK)/FLOAT(NOTYET)*(OVEREL-UNDSAV)
  272 I=I+1
      IM1= I-1
      IF(IM1.EQ.0) IM1=2


C       DO (AXIAL LOCATION INDICATORS)


C       PROCEDURE  (AXIAL LOCATION INDICATORS)

      INLET=.FALSE.
      EXIT  = .FALSE.
```

```
      IF(I.EQ.1) INLET =.TRUE.                                          QS008170
      IF( I.EQ. IEXIT) EXIT=.TRUE.                                      QS008180
C     END                                                               QS008190
275   J=J+1                                                             QS008200
      JM1=J-1                                                           QS008210
      IF(JM1.EQ.0) JM1=2                                                QS008220
                                                                        QS008230
                                                                        QS008240
C     DO (VERTICAL LOCATION INDICATORS)                                 QS008250
                                                                        QS008260
                                                                        QS008270
C     PROCEDURE (VERTICAL LOCATION INDICATORS)                          QS008280
                                                                        QS008290
      BOTTOM = .FALSE.                                                  QS008300
      TOP    = .FALSE.                                                  QS008310
      BLADE  = .FALSE.                                                  QS008320
      IF(J.EQ.1) BOTTOM =.TRUE.                                         QS008330
      IF(J.EQ.M) TOP =.TRUE.                                            QS008340
      BLADE = TOP                                                       QS008350
C     END                                                               QS008360
                                                                        QS008370
                                                                        QS008380
C     DO (BOUNDARY CONDITION SWITCHES)                                  QS008390
                                                                        QS008400
                                                                        QS008410
C      PROCEDURE (BOUNDARY CONDITION SWITCHES)                          QS008420
                                                                        QS008430
      DO 209 I1=1,8                                                     QS008440
209   ONOFF(I1)=1.                                                      QS008450
C     IF(INLET) THEN                                                    QS008460
C     ONOFF(3)= 0.                                                      QS0084[illegible]
C     ONOFF(4)= 0.                                                      QS00848[illegible]
C     ONOFF(5)= 0.                                                      QS008490
C     ONOFF(6)= 0.                                                      QS008500
C     END                                                               QS008510
C     IF(BOTTOM) THEN                                                   QS008520
C     ONOFF(5)= 0.                                                      QS008530
C     ONOFF(6)= 0                                                       QS008540
C     ONOFF(7)= 0.                                                      QS008550
C     ONOFF(8)= 0.                                                      QS008560
C     END                                                               QS008570
C     IF(TOP) THEN                                                      QS008580
C     ONOFF(1)= 0.                                                      QS008590
C     ONOFF(2)= 0.                                                      QS008600
C     ONOFF(3)= 0.                                                      QS008610
C     ONOFF(4)= 0.                                                      QS008620
C     END                                                               QS008630
      IF(.NOT.INLET.AND..NOT.BOTTOM.AND..NOT.TOP ) GOTO235              QS008640
```

```
      IF(TOP)IS=1
      IF(INLET)IS=3
      IF(BOTTOM)IS=5
      IE=IS+3
      DO 2099 I1=IS,IE
2099  ONOFF(I1)=0.
C     END


C     DO (GEOMETRY ADVANCE)

C     PROCEDURE (GEOMETRY ADVANCE)

235   HITE(1)=XDIF(HEIGH (I,J),HEIGHT(I+1,J),0.,0.)
      IF(I.EQ.IEXIT)HITE( )=HEIGHT(I,J)
      HITE(2)=XDIF(HEIGHT(I,J),HEIGHT(I,J+1),0.,0.)
      HITE(3)=HITE(2)
      HITE(8)=HITE(1)
      SPAN(1)=XDIF(RADIUS(I,J),RADIUS(I+1,J),0.,0.)
      IF(I.EQ.IEXIT)SPAN(1)=RADIUS(I,J)
      SPAN(1)=SPAN(1)/FIXR
      SPAN(2)=XDIF(RADIUS(I,J),RADIUS(I,J+1),0.,0.)
      SPAN(2)=SPAN(2)/FIXR
      SPAN(3)=SPAN(2)
      SPAN(8)=SPAN(1)
      T(6) = -T(3)*ONOFF(6)
      T(7) = -T(2)*ONOFF(7)

C     CALCULATE THE METRICS ON THE FIRST ITERATION
C     STORE THEM FOR SUBSEQUENT ITERATIONS

      IF(KK.GT.2)GOTO237

C     CALCULATE VELOCITIES AT UP AND DOWNSTREAM BOUNDARY ELEMENTS TO SAT
C     MASS FLOW
C         INDEX(I,J)>0 POINTS TO BOUNDARY ELEMENTS AFFECTED

      IF(INDEX(I,J).EQ.0.OR.KK.GT.1)GOTO2037
      ICOUNT=ICOUNT+1
      I1S=I1DEX(ICOUNT)
      IDIF=INDEX(I,J)+I1S-1
      DO 9999 I11=I1S,IDIF
      I1=I11
      IF(I1S.EQ.1.AND.I1.EQ.IDIF)I1=8
      ROTATX=ROTATN*SPAN(I1)*FIXR/RADLE
      BIGKH=BIGK/HITE(I1)/SPAN(I1)
```

```
      U0=U0OUT
      DPHDTH=DPHTHO
      IF(ICOUNT.GE.11.AND.ICOUNT.LE.24)U0=U0IN
      IF(ICOUNT.GE.11.AND.ICOUNT.LE.24)DPHDTH=DPHTHI
      FAKEV(IDEX)=DPHDTH/SPAN(I1)
      VREL2=(FAKEV(IDEX)-ROTATX)**2
      ROTATX=ROTATX*ROTATX-ROTATN*ROTATN
      SETTLE=.FALSE.
      YEW=U0
      IF(NOTYET.GT.KKMAX)YEW=BIGKH
      JCOUNT=0
 1999 QSQAR=YEW*YEW+VREL2
      IF(QSQAR.GE.QLIM)QSQAR=.99*QLIM
      RHOFAC=RHOCON*(1.-QSQAR)+1.
      RHOFAC=RHOFAC+ROTATX*RHOCON
      IF(RHOFAC.LT.0..OR.NOTYET.GT.KKMAX)RHOFAC=1.
      EFF=YEW-BIGKH*RHOFAC**(-EXPO)
      DFDU=1.-MINF*MINF*YEW*BIGKH*RHOFAC**(-GAMMA*EXPO)
      DU=-EFF/DFDU
      SETTLE=ABS(DU)/YEW.LT..001.OR.JCOUNT.GT.20
      YEW =YEW +DU
      JCOUNT=JCOUNT+1
      IF(.NOT.SETTLE) GOTO1999
      IF(JCOUNT.GT.20)YEW=BIGKH
      IF(JCOUNT.GT.20)WRITE(6,1677) I,J,I1
 1677 FORMAT('0CALCULATION OF VELOCITY AT BOUNDARY ELEMENT I= ',I3,
     1' J= ','I1= ',I3,' EXCEEDS 20 ITERATIONS. PROCEEDING WITH'/
     1 'INCOMPRESSIBLE ASSUMPTION (VELOCITY.EQ.MASS FLOW/AREA) ' )
      FAKEU(IDEX)=YEW
 9999 IDEX=IDEX+1
      ICOUNT=ICOUNT-1
      IDEX=IDEX-INDEX(I,J)



C     IF(KK.LE.2) THEN
 2037 IF(TOP) GOTO239
C     IF(.NOT.TOP) THEN
      XABMXA = XDIF(X(I+1,J+1),X(I,J+1),X(I,J),X(I+1,J))*.5
      XABMXB =.5*XDIF(X(I+1,J+1),X(I+1,J),X(I,J),X(I,J+1))
      YABMYA =.5*XDIF(Y(I+1,J+1),Y(I,J+1),Y(I,J),Y(I+1,J))
      YABMYB =.5*XDIF(Y(I+1,J+1),Y(I+1,J),Y(I,J),Y(I,J+1))
      XBCMXB =.5*XDIF(X(IM1,J+1),X(IM1,J),X(I,J),X(I,J+1))
      YBCMYB =.5*XDIF(Y(IM1,J+1),Y(IM1,J),Y(I,J),Y(I,J+1))
      LAB    = SQRT( XABMXA*XABMXA + SPAN(1)*YABMYA*SPAN(1)*YABMYA)
      HAB    = SQRT( XABMXB*XABMXB + SPAN(2)*YABMYB*SPAN(2)*YABMYB)
      HBC    = SQRT( XBCMXB*XBCMXB + SPAN(3)*YBCMYB*SPAN(3)*YBCMYB)
      COSX(1)=  YABMYA*SPAN(1)/LAB
```

```
      COSX(2)=     -YABMYB*SPAN(2)/HAB                                    QS009610
      COSX(3)=     YBC4YB*SPAN(3)/HBC                                     QS009620
      COSY(1)=                -XABMXA/LAB                                 QS009630
      COSY(2)=   XABMXB/HAB                                               QS009640
      COSY(3)= -XBCMXB/HBC                                                QS009650
C     END                                                                 QS009660
239   IF(BOTTOM)GOTO241                                                   QS009670
C     IF(.NOT.BOTTOM) THEN                                                QS009680
      XADMXA =.5*XDIF(X(I+1,JM1),X(I,JM1),X(I,J),X(I+1,J))                QS009690
      YADMYA =.5*XDIF(Y(I+1,JM1),Y(I,JM1),Y(I,J),Y(I+1,J))                QS009700
      LAD    = SQRT(XADMXA*XADMXA + SPAN(8)*YADMYA*SPAN(8)*YADMYA)        QS009710
      COSX(8)= -YADMYA*SPAN(8)/LAD                                        QS009720
      COSY(8)=  XADMXA/LAD                                                QS009730
C     END                                                                 QS009740
241   LENGTH(1) =  LAB                                                    QS009750
      LENGTH(2) =  HAB                                                    QS009760
      LENGTH(3) =  HBC                                                    QS009770
      LENGTH(8) =  LAD                                                    QS009780
      EXDIF1=X(I+1,J)-X(I,J)                                              QS009790
      EXDIF2=X(IM1,J)-X(I,J)                                              QS009800
      YDIF1=SPAN(1)*(Y(I+1,J)-Y(I,J))                                     QS009810
      YDIF2=Y(IM1,J)-Y(I,J)                                               QS009820
      SLEN=EXDIF1*EXDIF1+YDIF1*YDIF1                                      QS009830
      ZETDF1     = ZETA(J+1)  - ZETA(J)                                   QS009840
      ZETDF2     = ZETA(JM1)  - ZETA(J)                                   QS009850
      ETADF1     = ETA(I+1)   - ETA(I)                                    QS009860
      ETADF2 = ETA(IM1)-ETA(I)                                            QS009870
C     CELL I, POINT  A                                                    QS009880
      EK4 = 2.*YABMYA/ZETDF1                                              QS009890
      EK3 = (Y(I+1,J)-Y(I,J))/ETADF1                                      QS009900
      EK2 = 2.*XABMXA/ZETDF1                                              QS009910
      EK1 = (X(I+1,J)-X(I,J))/ETADF1                                      QS009920
      L = 1                                                               QS009930
C     DO (TRANSFORMATION MATRIX)                                          QS009940
C     PROCEDURE (TRANSFORMATION MATRIX)                                   QS009950
      JACOB=EK1*EK4-EK2*EK3                                               QS009960
      IF(JACOB.EQ.0.) GOTO243                                             QS009970
C     IF(JACOB.NE.0.) THEN                                                QS009980
      DEDX(L)= EK4/JACOB                                                  QS009990
      DEDY(L)=-EK2/JACOB/SPAN(1)                                          QS010000
      DZDX(L)= -EK3/JACOB                                                 QS010010
      DZDY(L)=  EK1/JACOB/SPAN(1)                                         QS010020
C     END                                                                 QS010030
C     END                                                                 QS010040
243   EK4 = 2.*YADMYA/ZETDF2                                              QS010050
C           CELL  IV,  POINT A                                            QS010060
      EK2 = 2.*XADMXA/ZETDF2                                              QS010070
      L=8                                                                 QS010080
```

```
C     DO (TRANSFORMATION MATRIX)                                        QS010090
      JACOB=EK1*EK4-EK2*EK3                                             QS010100
      IF(JACOB.EQ.0.)GOTO245                                            QS010110
C     IF(JACOB.NE.0.) THEN                                              QS010120
      DEDX(L)= EK4/JACOB                                                QS010130
      DEDY(L)=-EK2/JACOB/SPAN(8)                                        QS010140
      DZDX(L)= -EK3/JACOB                                               QS010150
      DZDY(L)=  EK1/JACOB/SPAN(8)                                       QS010160
C     END                                                               QS010170
C     CELL II, POINT B                                                  QS010180
245   EK4  = (Y(I,J+1)-Y(I,J))/ZETDF1                                   QS010190
      EK3  = 2.*YBCMYB/ETADF2                                           QS010200
      EK2  = (X(I,J+1)-X(I,J))/ZETDF1                                   QS010210
      EK1  = 2.*XBCMXB/ETADF2                                           QS010220
      L=3                                                               QS010230
C     DO (TRANSFORMATION MATRIX)                                        QS010240
      JACOB=EK1*EK4-EK2*EK3                                             QS010250
      IF(JACOB.EQ.0.)GOTO247                                            QS010260
C     IF(JACOB.NE.0.) THEN                                              QS010270
      DEDX(L)= EK4/JACOB                                                QS010280
      DEDY(L)=-EK2/JACOB/SPAN(3)                                        QS010290
      DZDX(L)= -EK3/JACOB                                               QS010300
      DZDY(L)=  EK1/JACOB/SPAN(3)                                       QS010310
C     END                                                               QS010320
C     CELL I, POINT B                                                   QS010330
247   EK1 = 2.*XABMXB/ETADF1                                            QS010340
      EK3 = 2.*YABMYB/ETADF1                                            QS010350
      L=2                                                               QS010360
C     DO (TRANSFORMATION MATRIX)                                        QS010370
      JACOB=EK1*EK4-EK2*EK3                                             QS010380
      IF(JACOB.EQ.0.)GOTO249                                            QS010390
C     IF(JACOB.NE.0.) THEN                                              QS010400
      DEDX(L)= EK4/JACOB                                                QS010410
      DEDY(L)=-EK2/JACOB/SPAN(2)                                        QS010420
      DZDX(L)= -EK3/JACOB                                               QS010430
      DZDY(L)=  EK1/JACOB/SPAN(2)                                       QS010440
C     END                                                               QS010450
249   T(1)=LENGTH(1)*HITE(1)*(DEDX(1)*COSX(1)+DEDY(1)*COSY(1            QS010460
     1))*ONOFF(1)                                                       QS010470
      T(2)=LENGTH(2)*HITE(2)*(DZDX(2)*COSX(2)+DZDY(2)*COSY(2            QS010480
     1))*ONOFF(2)                                                       QS010490
      T(3)=LENGTH(3)*HITE(3)*(DZDX(3)*COSX(3)+DZDY(3)*COSY(3            QS010500
     1))*ONOFF(3)                                                       QS010510
      T(8)=LENGTH(8)*HITE(8)*(DEDX(8)*COSX(8)+DEDY(8)*COSY(8            QS010520
     1))*ONOFF(8)                                                       QS010530
                                                                        QS010540
C     STORE METRICS AND GEOMETRY AT POINT I,J                           QS010550
                                                                        QS010560
```

```
      SUPERG(1,J,I)=COSX(1)                                             QS010570
      SUPERG(2,J,I)=COSX(2)                                             QS010580
      SUPERG(3,J,I)=COSX(3)                                             QS010590
      SUPERG(4,J,I)=COSX(8)                                             QS010600
      SUPERG(5,J,I)=COSY(1)                                             QS010610
      SUPERG(6,J,I)=COSY(2)                                             QS010620
      SUPERG(7,J,I)=COSY(3)                                             QS010630
      SUPERG(8,J,I)=COSY(8)                                             QS010640
      SUPERG(9,J,I)=LENGTH(1)                                           QS010650
      SUPERG(10,J,I)=LENGTH(2)                                          QS010660
      SUPERG(11,J,I)=LENGTH(3)                                          QS010670
      SUPERG(12,J,I)=LENGTH(8)                                          QS010680
      SUPERG(13,J,I)=EXDIF1                                             QS010690
      SUPERG(14,J,I)=EXDIF2                                             QS010700
      SUPERG(15,J,I)=YDIF1                                              QS010710
      SUPERG(16,J,I)=YDIF2                                              QS010720
      SUPERG(17,J,I)=SLEN                                               QS010730
      SUPERG(18,J,I)=SLEN2                                              QS010740
      SUPERG(19,J,I)=ZETDF1                                             QS010750
      SUPERG(20,J,I)=ZETDF2                                             QS010760
      SUPERG(21,J,I)=ETADF1                                             QS010770
      SUPERG(22,J,I)=ETADF2                                             QS010780
      SUPERG(23,J,I)=DEDX(1)                                            QS010790
      SUPERG(24,J,I)=DEDX(2)                                            QS010800
      SUPERG(25,J,I)=DEDX(3)                                            QS010810
      SUPERG(26,J,I)=DEDX(8)                                            QS010820
      SUPERG(27,J,I)=DEDY(1)                                            QS010830
      SUPERG(28,J,I)=DEDY(2)                                            QS010840
      SUPERG(29,J,I)=DEDY(3)                                            QS010850
      SUPERG(30,J,I)=DEDY(8)                                            QS010860
      SUPERG(31,J,I)=DZDX(1)                                            QS010870
      SUPERG(32,J,I)=DZDX(2)                                            QS010880
      SUPERG(33,J,I)=DZDX(3)                                            QS010890
      SUPERG(34,J,I)=DZDX(8)                                            QS010900
      SUPERG(35,J,I)=DZDY(1)                                            QS010910
      SUPERG(36,J,I)=DZDY(2)                                            QS010920
      SUPERG(37,J,I)=DZDY(3)                                            QS010930
      SUPERG(38,J,I)=DZDY(8)                                            QS010940
      SUPERG(39,J,I)=T(1)                                               QS010950
      SUPERG(40,J,I)=T(2)                                               QS010960
      SUPERG(41,J,I)=T(3)                                               QS010970
      SUPERG(42,J,I)=T(8)                                               QS010980
C       SINCE( KK.LE.2)                                                 QS010990
      GOTO 238                                                          QS011000
                                                                        QS011010
C     ELSE                                                              QS011020
                                                                        QS011030
C    IF ITERATION COUNT EXCEEDS 2, OBTAIN METRICS, ETC. FROM STORAGE    QS011040
```

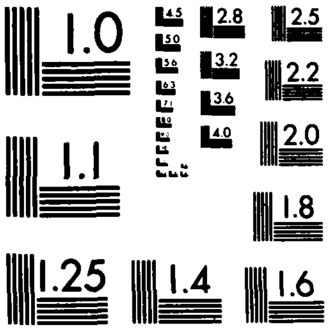
AD-A120 456 PRELIMINARY MEASUREMENTS AND CODE CALCULATIONS OF FLOW THROUGH A CASCADE OF DCA BLADING AT A SOLIDITY OF 167 (U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W D MOLLOY JUN 82 3/3

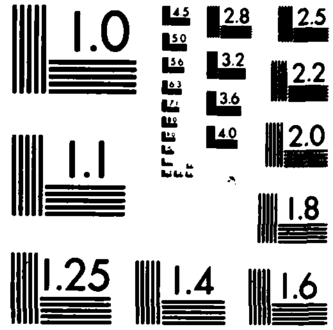
UNCLASSIFIED F/G 21/5 NL

END

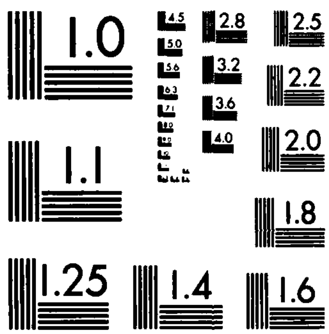
FILMED

DTIC

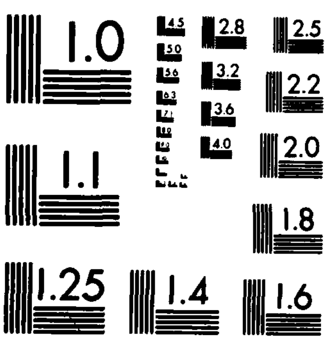




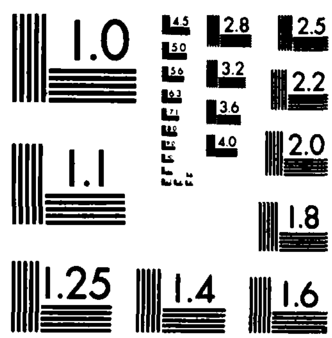
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
                                                                QS011050
237   COSX(1)=SUPERG(1,J,I)                                     QS011060
      COSX(2)=SUPERG(2,J,I)                                     QS011070
      COSX(3)=SUPERG(3,J,I)                                     QS011080
      COSX(8)=SUPERG(4,J,I)                                     QS011090
      COSY(1)=SUPERG(5,J,I)                                     QS011100
      COSY(2)=SUPERG(6,J,I)                                     QS011110
      COSY(3)=SUPERG(7,J,I)                                     QS011120
      COSY(8)=SUPERG(8,J,I)                                     QS011130
      LENGTH(1)=SUPERG(9,J,I)                                   QS011140
      LENGTH(2)=SUPERG(10,J,I)                                  QS011150
      LENGTH(3)=SUPERG(11,J,I)                                  QS011160
      LENGTH(8)=SUPERG(12,J,I)                                  QS011170
      EXDIF1=SUPERG(13,J,I)                                     QS011180
      EXDIF2=SUPERG(14,J,I)                                     QS011190
      YDIF1=SUPERG(15,J,I)                                      QS011200
      YDIF2=SUPERG(16,J,I)                                      QS011210
      SLEN=SUPERG(17,J,I)                                       QS011220
      SLEN2=SUPERG(18,J,I)                                      QS011230
      ZETDF1=SUPERG(19,J,I)                                     QS011240
      ZETDF2=SUPERG(20,J,I)                                     QS011250
      ETADF1=SUPERG(21,J,I)                                     QS011260
      ETADF2=SUPERG(22,J,I)                                     QS011270
      DEDX(1)=SUPERG(23,J,I)                                    QS011280
      DEDX(2)=SUPERG(24,J,I)                                    QS011290
      DEDX(3)=SUPERG(25,J,I)                                    QS011300
      DEDX(8)=SUPERG(26,J,I)                                    QS011310
      DEDY(1)=SUPERG(27,J,I)                                    QS011320
      DEDY(2)=SUPERG(28,J,I)                                    QS011330
      DEDY(3)=SUPERG(29,J,I)                                    QS011340
      DEDY(8)=SUPERG(30,J,I)                                    QS011350
      DZDX(1)=SUPERG(31,J,I)                                    QS011360
      DZDX(2)=SUPERG(32,J,I)                                    QS011370
      DZDX(3)=SUPERG(33,J,I)                                    QS011380
      DZDX(8)=SUPERG(34,J,I)                                    QS011390
      DZDY(1)=SUPERG(35,J,I)                                    QS011400
      DZDY(2)=SUPERG(36,J,I)                                    QS011410
      DZDY(3)=SUPERG(37,J,I)                                    QS011420
      DZDY(8)=SUPERG(38,J,I)                                    QS011430
      T(1)=SUPERG(39,J,I)                                       QS011440
      T(2)=SUPERG(40,J,I)                                       QS011450
      T(3)=SUPERG(41,J,I)                                       QS011460
      T(8)=SUPERG(42,J,I)                                       QS011470
C     END                                                       QS011480
C     CELL  II,  POINT  C                                       QS011490
238   T(5) =   PRIORS(1,J)*ONOFF(5)                             QS011500
      T(4) =   PRIORS(2,J)*ONOFF(4)                             QS011510
      PRIORS(1,J) = - T(8)                                      QS011520
```

```
      PRIORS(2,J) = - T(1)
C     END


C     DO (FIELD GRADIENTS)

C     PROCEDURE (FIELD GRADIENTS)

      DO 438 I1=1,3
      FI1(I1)=PHI(I+1,J)
      FI2(I1)=PHI(I+1,J+1)
      FI3(I')=PHI(I  ,J+1)
438   FI9(I1)=PHI(I  ,J  )
      FI4    =PHI(IM1,J+1)
      FI5    =PHI(IM1,J  )
      FI6    =PHI(IM1,JM1)
      FI7    =PHI(I  ,JM1)
      FI8    =PHI(I+1,JM1)
      FI1(8)=PHI(I+1,J)
      FI9(8)=PHI(I  ,J)
      I10=I-IEXIT+3
C     CELL I , POINT A
251   DPHIDZ(1)= XDIF(FI3(1),FI2(1),FI9(1),FI1(1))/ZETDF1
      DPHIDE(1) =  (FI1(1)-FI9(1))/ETADF1
C     CELL I, POINT B
      DPHIDE(2)=XDIF(FI1(2),FI2(2),FI9(2),FI3(2)) /ETADF1
      DPHIDZ(2)=(FI3(2)-FI9(2))/ZETDF1
C     CELL II, POINT B
      DPHIDE(3)=XDIF(FI5,FI4,FI9(3),FI3(3)) /ETADF2
      DPHIDZ(3)=(FI3(3)-FI9(3))/ZETDF1
C     CELL IV, POINT A
      DPHIDE(8) =  (FI1(8)-FI9(8))/ETADF1
      DPHIDZ(8) =XDIF(FI7,FI8,FI9(8),FI1(8))/ZETDF2
C     END


C     DO (VELOCITIES AND DENSITIES)

C     PROCEDURE (VELOCITIES AND DENSITIES)

      U(6) = U(3)
      V(6) = V(3)
      U(7) = U(2)
```

```
      V(7) = V(2)
      DO 541 I1 =1,8
      IF(I1.GE.4.AND.I1.NE.8) GOTO 541
C     IF(I1.LT.4.OR.I1.EQ.8) THEN
      U(I1)=DPHIDE(I1)*DEDX(I1)+ DPHIDZ(I1)*DZDX(I1)
      V(I1)=DPHIDE(I1)*DEDY(I1)+ DPHIDZ(I1)*DZDY(I1)
      IF(INDEX(I,J).EQ.0)GOTO5411
      ICOUNT=ICOUNT+1
      I1S=I1DEX(ICOUNT)
      IF(I1.LE.I1S)I1P=I1S
      IDIF=INDEX(I,J)+I1S-1
      IF(I1S.EQ.1.AND.I1P.EQ.IDIF)I1P=8
      IF(I1.NE.I1P)GOTO5410
      IF(I1P.GT.IDIF.AND.I1P.NE.8)GOTO5410
      V(I1)=FAKEV(IDEX)
      U(I1)=FAKEU(IDEX)
      IDEX=IDEX+1
      I1P=I1P+1
5410  IF(I1.NE.8)ICOUNT=ICOUNT-1
5411  V(I1)=V(I1)-SPAN(I1)*FIXR/RADLE*ROTATN
      UN(I1)=U(I1)*COSX(I1)+V(I1)*COSY(I1)
C     END
541   CONTINUE
      IF(.NOT.TOP) GOTO 2003
      U8=SQRT(U(8)*U(8)+V(8)*V(8))*ABS(EXDIF1)/SQRT(SLEN)
      IF(KK.GT.7.OR.ABS(FLOCO).GT..01)U(8)=SIGN(U8,U(8))
      V(8)=U(8)*YDIF1/EXDIF1
      UN(8)=U(8)*COSX(8)+V(8)*COSY(8)
C     END
2003  U(4) =  PRIORS(3,J)
      U(5) =  PRIORS(4,J)
      V(4) =  PRIORS(5,J)
      V(5) =  PRIORS(6,J)
      PRIORS(3,J)=U(1)
      PRIORS(4,J)=U(8)
      PRIORS(5,J)=V(1)
      PRIORS(6,J)=V(8)
      RHO(7)=RHO(2)
      RHO(6)=RHO(3)
      RHO(4)=PRIORS(7,J)
      RHO(5)=PRIORS(8,J)
      ONSUM=0.
      DO 595 I1=1,8

C     IF(KK.LE.2.OR.MOD(KK,2).EQ.0) THEN
      IF(I1.GE.4.AND.I1.NE.8) GOTO 2005
C     IF(I1.LT.4.OR.I1.EQ.8) THEN
      RHOFAC=1.
```

```
      RHO(I1)=1.
      IF((I.EQ.2.OR.I.EQ.IEXIT).AND.J.LE.4.AND.ROTATN.NE.0.
     1.OR.KK.LE.NOTYET) GOTO 2005
      QSQAR=U(I1)*U(I1)+(V(I1)*V(I1))
      IF(QSQAR.GE.QLIM)QSQAR=.99*QLIM
      RHOFAC=RHOCON*(1.-QSQAR)*ONOFF(I1)+1.
      RHO(I1)=1.-AMZ*(QSQAR-1.)+AMY*(QSQAR-1.)*(QSQAR-1.)
      IF(QSQAR.LT..65.OR.QSQAR.GT.1.44)RHO(I1)=RHOFAC**EXPO

C CORRECT DENSITY FOR ROTATION

      ROTATX=SPAN(I1)*FIXR/RADLE
      IF(ROTROT.NE.0..AND.J.GT.3)
     1RHO(I1)=(RHO(I1)**(1./EXPO)+RHOCON*ROTROT*((ROTATX)
     1**2-1.)*ONOFF(I1))**EXPO
2005  QSQAR=U(I1)*U(I1)+V(I1)*V(I1)
      IF(I.GT.1)
     1MACH(I1)=MINF*SQRT(QSQAR)/RHO(I1)**(1./EXPO)
C    1MACH(I1)=MINF*SQRT(QSQAR/RHO(I1)**(1./EXPO))
      IF(MACH(I1).GT.2.)MACH(I1)=2.
595   CONTINUE
      PRIORS(7,J)=RHO(1)
      PRIORS(8,J)=RHO(8)
C     END


C     DO (SUPERSONIC CORRECTION)


C     PROCEDURE (SUPERSONIC CORRECTION)

      ONSUM=0.
      ROSUM=RHO(4)+RHO(5)+RHO(6)+RHO(7)
      EMACH(I,J)=0.
      XVELIJ=0.
      DO 608 I3=1,8
      RHOSS(I3)=0.
608   ONSUM=ONSUM+ONOFF(I3)
      YVELIJ=0.
      SMALLA=0.
      SMALLB=0.
      ZETDF1=ZETA(J+1)-ZETA(J)
      IF(J.EQ.M)ZETDF1=ZETA(J)-ZETA(J-1)
      EMACH(I,J)=(ONOFF(1)*MACH(1)+ONOFF(8)*MACH(8))/(ONOFF(1)+ONOFF(8)
     1)
      EMACH(IEXIT+1,J)=EMACH(3,J)
      EMACH(1,J)=EMACH(IEXIT-1,J)
```

```
C     IF(KK.LE.2.OR.MOD(KK,IT).EQ.0.OR.MOD(KK,IT).GE.IT/2) THEN         QS012970
C       STORE DENSITIES FROM SURFACE OF ELEMENT IN DUMMY ARRAYS         QS012980
                                                                        QS012990
      DUMBX(I,J)=(ONOFF(1)*RHO(1)+ONOFF(8)*RHO(8))/(ONOFF(1)+ONOFF(8)   QS013000
     1)                                                                 QS013010
      IF(.NOT.TOP)DUMBY(I,J)=(ONOFF(2)*RHO(2)+ONOFF(3)*RHO(3))/         QS013020
     1(ONOFF(2)+ONOFF(3))                                               QS013030
      IF(.NOT.TOP)ZMACH(I,J)=(ONOFF(2)*MACH(2)+ONOFF(3)*MACH(3))/       QS013040
     1(ONOFF(2)+ONOFF(3))                                               QS013050
      IF(TOP)DUMBY(I,J)=2.*DUMBY(I,J-1)-DUMBY(I,J-2)                    QS013060
      IF(TOP)ZMACH(I,J)=2.*ZMACH(I,J-1)-ZMACH(I,J-2)                    QS013070
      DUMBX(IEXIT+1,J)=DUMBX(3,J)                                       QS013080
      ZMACH(1,J)=ZMACH(IEXIT-1,J)                                       QS013090
      DUMBY(I,M+1)=2.*DUMBY(I,1)-DUMBY(I,2)                             QS013100
      ZMACH(I,M+1)=2.*ZMACH(I,1)-ZMACH(I,2)                             QS013110
                                                                        QS013120
C     CALCULATE DENSITY CORRECTION TERMS,  MU*DRHODS*DS                 QS013130
                                                                        QS013140
C     IF(J.GT.5.OR.I.LT.NOZ-1.OR.I.GT.NOZ+1) THEN                       QS013150
      IF(J.LE.5.AND.I.GE.NOZ-1.AND.I.LE.NOZ+1) GOTO20001                QS013160
      DO 694 I1=2,8,6                                                   QS013170
      IF(TOP.AND.I1.NE.8) GOTO 694                                      QS013180
C     IF(.NOT.TOP.OR.I1.EQ.8) THEN                                      QS013190
      I1P1=I1+1                                                         QS013200
      IF(I1.EQ.8)I1P1=1                                                 QS013210
      ONSUM=ONOFF(I1)+ONOFF(I1P1)                                       QS013220
      XVELIJ=U(I1)*ONOFF(I1)/ONSUM                                      QS013230
      XVELIJ=XVELIJ+U(I1P1)*ONOFF(I1P1)/ONSUM                           QS013240
      YVELIJ=V(I1)*ONOFF(I1)/ONSUM                                      QS013250
      YVELIJ=YVELIJ+V(I1P1)*ONOFF(I1P1)/ONSUM                           QS013260
      SMALLA=(U(I1)*DZDX(I1)+V(I1)*DZDY(I1))*ONOFF(I1)/ONSUM            QS013270
      SMALLA=SMALLA+(U(I1P1)*DZDX(I1P1)+V(I1P1)*DZDY(I1P1))             QS013280
     1*ONOFF(I1P1)/ONSUM                                                QS013290
      SMALLB=(U(I1)*DEDX(I1)+V(I1)*DEDY(I1))*ONOFF(I1)                  QS013300
     1/ONSUM                                                            QS013310
      SMALLB=SMALLB+(U(I1P1)*DEDX(I1P1)+V(I1P1)*DEDY(I1P1))             QS013320
     1*ONOFF(I1P1)/ONSUM                                                QS013330
      IF(KK.LE.NOTYET) GOTO694                                          QS013340
C     IF(KK.GT.NOTYET) THEN                                             QS013350
      IF(BLADE)SMALLA=0.                                                QS013360
      SIGNOA=SIGN(1.,SMALLA)                                            QS013370
      IF(SMALLA.EQ.0.)SIGNOA=1.                                         QS013380
      FLOWAT=AMAX1(0.,SIGNOA)                                           QS013390
      SIGNOB=SIGN(1.,SMALLB)                                            QS013400
      FLOWBY=AMAX1(0.,SIGNOB)                                           QS013410
      QUE=SQRT(XVELIJ**2+YVELIJ**2)                                     QS013420
C        SLP2=TEGARD                                                    QS013430
      IF(ABS(SMALLB)/QUE.GT.SLP2)QUE=SQRT(SMALLA**2+SMALLB**2)          QS013440
```

```
      MYJ=J-IFIX(SIGNOA)
      MYI=I-IFIX(SIGNOB)
      JYM=MYJ+IFIX(FLOWAT)
      IYM=MYI+IFIX(FLOWBY)
      IF(MYJ.EQ.0)MYJ=2
      IF(MYI.EQ.0)MYI=IEXIT-2
      IYM1=IYM-1
      IF(IYM1.EQ.0)IYM1=1
      JYM1=JYM-1
      IF(JYM1.EQ.0)JYM1=1
      IF(I1.EQ.2) GOTO2019
C     IF(I1.EQ.8) THEN
      MACHA=(EMACH(I,J)+EMACH(I,MYJ))/2.
      MACHB=(EMACH(I,J)+EMACH(MYI,J))/2.
      SWICHA=-AMAX1(0.,1./SHIFTM**2-1./MACHA/MACHA)/QUE
      SWICHB=-AMAX1(0.,1./SHIFTM**2-1./MACHB/MACHB)/QUE

C    RHOSS(2)=MU*DRHODS*DS AT TOP OF ELEMENT
C    RHOSS(8)=MU*DRHODS*DS ON FORWARD, SWEEPING EDGE OF ELEMENT
      RHOSS(I1)=SWICHB*SIGNOB*SMALLB*(DUMBX(I,J)-DUMBX(MYI,J))+
     1          SWICHA*SIGNOA*SMALLA*(DUMBX(I,J)-DUMBX(I,MYJ))
C     SINCE (I1.NE.2)
      GOTO 2020
2019  MACHA=(ZMACH(I,J)+ZMACH(I,MYJ))/2.
      MACHB=(ZMACH(I,J)+ZMACH(MYI,J))/2.
      SWICHA=-AMAX1(0.,1./SHIFTM**2-1./MACHA/MACHA)/QUE
      SWICHB=-AMAX1(0.,1./SHIFTM**2-1./MACHB/MACHB)/QUE
      RHOSS(I1)=SWICHB*SIGNOB*SMALLB*(DUMBY(I,J)-DUMBY(MYI,J))+
     1          SWICHA*SIGNOA*SMALLA*(DUMBY(I,J)-DUMBY(I,MYJ))
C     END
2020  RHOSS(I1P1)=RHOSS(I1)
      RHO(I1)=RHO(I1)+DAMP*RHOSS(I1)*ONOFF(I1)
      IF(RHO(I1).LE.0.)RHO(I1)=RHO(I1)-2.*
     1DAMP*RHOSS(I1)*ONOFF(I1)
      RHO(I1P1)=RHO(I1P1)+DAMP*RHOSS(I1P1)*ONOFF(I1P1)
      IF(RHO(I1P1).LE.0.)RHO(I1P1)=RHO(I1P1)-2.*
     1DAMP*RHOSS(I1P1)*ONOFF(I1P1)
      PRIORS(7,J)=RHO(1)
      PRIORS(8,J)=RHO(8)
3334  FORMAT(2I8,7E13.5)
C     END
C     END
694   CONTINUE
C     END
20001 CONTINUE
C     END
C     END
```

```

C       DO ( T'S,Q'S AND RESIDUALS)


C       PROCEDURE ( T'S,Q'S AND RESIDUALS)

C       NET MASS FLOW TERM
        ONSUM=0.
        FLONET = -RHOUL(2)*ONOFF(7)-RHOUL(3)*ONOFF(6)+PRIORS(9,J)*ONOFF(4
       1) + PRIORS(10,J)*ONOFF(5)
        DO 711 I1=1,8
        ONSUM=ONSUM+ONOFF(I1)
        IF(I1.GE.4.AND.I1.NE.8) GOTO 711
C       IF(I1   .LT.4.OR.I1.EQ.8) THEN
        ROSUM=ROSUM+RHO(I1)
        RHOUL(I1)= RHO(I1)*UN(I1)*LENGTH(I1)* HITE(I1)*ONOFF(I1)
        FLONET =FLONET + RHOUL(I1)
C       END
711     CONTINUE
        ROAVG=ROSUM/ONSUM
        TENOLF(I,J)=FLONET
        PRIORS(9,J) = -RHOUL(1)
        PRIORS(10,J) = -RHOUL(8)
C
C       RESIDUAL TOTAL
        R=-FLONET
C       END
C       IF(BUG) DO (FIRST DEBUG LIST)


C       PROCEDURE (FIRST DEBUG LIST)

C        BUG AND BUG3 ARE NOT INCLUDED IN NAMELIST PARAMS. THEY MUST BE
C         ADDED OR SET HERE MANUALLY IN ORDER TO ACTIVATE THESE PRINTOUTS.

        IF(.NOT.EXIT.OR.MOD(KK,IT).NE.0.OR..NOT.TOP.OR..NOT.BUG) GOTO 2023
C       IF(EXIT.AND.MOD(KK,IT).EQ.0.AND.TOP.AND.BUG) THEN
        WRITE(6,YTESD)
        IF(.NOT.BUG3) GOTO2025
C       IF(BUG3) THEN
        WRITE(6,7001)
7001    FORMAT('OARRAY OF POTENTIAL CORRECTIONS,BLADE SURFACE=FAR RIGHT',
       1' COLUMN')
        DO 732 I2=1,IEXIT
732     WRITE(6,9)(DELPSV(I2,J2),J2=I67,M)
        WRITE(6,7002)
7002    FORMAT('OARRAY OF POTENTIAL VALUES,BLADE SURFACE=FAR RIGHT ',
```

```
     1'COLUMN')
      DO 738 I2=1,IEXIT
738   WRITE(6,9)(PHI(I2,J2),J2=I67,M)
C     END
2025  WRITE(6,7003)
7003  FORMAT('OARRAY OF RESIDUAL MASS FLOWS,BLADE SURFACE=FAR RIGHT',
     1' COLUMN')
      DO 745 I2=1,IEXIT
745   WRITE(6,9)(TENOLF(I2,J2),J2=I67,M)
      WRITE(6,7004)
7004  FORMAT('OARRAY OF MACH NUMBERS,BLADE SURFACE=FAR RIGHT COLUMN')
      DO 750 I2=1,IEXIT
750   WRITE(6,9)(EMACH(I2,J2),J2=I67,M)
      REWIND 8
      WRITE(8,2) PHI
C     END
9     FORMAT(1H ,14E9.3)
C     END


C     DO ( DELPHI COEFFICIENTS AND TRANSPOSED TERMS)

C     PROCEDURE ( DELPHI COEFFICIENTS AND TRANSPOSED TERMS)
C
C     AIJP1*DELPHI(I,J+1)+AIJ*DELPHI(I,J)+AIJM1*DELPHI(I,J-1)= R + BIGZ

C
2023  DELPM1(2) = DELPHI(J)
      IF(I.EQ.2)DELPM1(2)=DELPSV(IEXIT-1,J)
      IF(KK.GT.2.AND.MOD(KK,IT).NE.0) GOTO 2029
C     IF(KK.LE.2.OR.MOD(KK,IT).EQ.0) THEN
      AIJP1=(T(2)*RHO(2)+T(3)*RHO(3))/ZETDF1
      AIJM1=(T(6)*RHO(6)+T(7)*RHO(7))/ZETDF2
      TT18=(T(1)*RHO(1)+T(8)*RHO(8))/ETADF1
      ZIM1J=(T(4)*RHO(4)+T(5)*RHO(5))/ETADF2
      AIJ=-1.*(AIJM1+AIJP1+TT18+ZIM1J)
      ZIP1J=0.
      ZJP1=0.
      ZJM1=0.

C    SAVE COEFFICIENTS UNTIL THE NEXT TIME DENSITY IS UPDATED (WHEN KK IS
C    DIVISIBLE BY IT).
      SUPERG(43,J,I)=AIJ
      SUPERG(44,J,I)=AIJM1
      SUPERG(45,J,I)=AIJP1
      SUPERG(46,J,I)=ZIM1J
      ZIM1J=ZIM1J*DELPM1(2)
```

```
C       SINCE (KK.LE.2.OR.MOD(KK,IT).EQ.0)
        GOTO 2030

C     IF ITERATION COUNT EXCEEDS 2 AND IS NOT DIVISIBLE BY IT, OBTAIN
C     COEFFICIENTS FROM THE LAST CALCULATION.
2029    AIJ=SUPERG(43,J,I)
        AIJM1=SUPERG(44,J,I)
        AIJP1=SUPERG(45,J,I)
        ZIM1J=SUPERG(46,J,I)
        ZIM1J=ZIM1J*DELPM1(2)
        ZJP1=0.
        ZJM1=0.
C       END
2030    BIGZ = -(ZIP1J +ZIM1J + ZJP1 + ZJM1)
        IF(J.EQ.4)BIGZ=BIGZ-AIJM1*DELPSV(IEXIT+2-I,5)
        TRANSP = R +BIGZ

C      ELIMINATE CHANGES TO PHI(BY ZEROING COEFFICIENTS)ON ALL DUMMY LINES
C      BEYOND  PERIODIC LINE.
        IF(.NOT.(J.LT.4.OR.(J.LE.5.AND.
       1(I.LE.3.OR.I.GE.IEXIT-1.OR.I.EQ.NOZ-1.OR.
       1I.EQ.NOZ.OR.I.EQ.NOZ+1)))) GOTO 2031
C       AIJ=0.
        AIJM1=0.
        AIJP1=0.
        TRANSP=0.

C       STORE COEFFICIENTS FOR LATER TRIDIAGONAL SOLUTION OF THIS LINE
2031    PRIORS(11,J)=AIJ
        PRIORS(12,J)=AIJP1
        PRIORS(13,J)=AIJM1
        PRIORS(14,J)=TRANSP


C     PROCEED TO NEXT VALUE OF J ON THIS RADIATING LINE
        IF(.NOT.TOP)GOTO275
C       END



C       DO (FORWARD ELIMINATION)


C       PROCEDURE (FORWARD ELIMINATION)

        DO 828 J=1,M
        JM1=J-1
        IF(JM1.EQ.0) JM1=2
```

```
      BOTTOM = .FALSE.
      TOP    = .FALSE.
      BLADE  = .FALSE.
      IF(J.EQ.1) BOTTOM =.TRUE.
      IF(J.EQ.M) TOP =.TRUE.
      BLADE = TOP

C    RECOVER COEFFICIENTS
      AIJ=PRIORS(11,J)
      AIJP1=PRIORS(12,J)
      AIJM1=PRIORS(13,J)
      TRANSP=PRIORS(14,J)
      IF(BOTTOM) GOTO 828
C     IF(.NOT.BOTTOM) THEN
      FFAC=AIJ-AIJM1*E(JM1)
      E(J)= +AIJP1/FFAC
      F(J)=  (TRANSP - AIJM1*F(JM1))/FFAC
3333  FORMAT(9E13.5)
C     END

828   CONTINUE
C     END


C     IF(I.NE.1)
C    1DO (BACK SUBSTITUTION)


C     PROCEDURE (BACK SUBSTITUTION)

      DELPHI(M)=F(M)
      OV=OVEREL
      IF(KKK.LT.NOTYET)OV=UNDERL
      IF(EMACH(I,M).GE.1.)OV=SUPREL*(1.-D(M)*(1.-CI))
      DELPSV(I,M)=DELPHI(M)*OV
      J=M-1
856   IF(J.LT.1)GOTO857
3335  FORMAT(3I6,9E12.5)
      BOTTOM = .FALSE.
      TOP    = .FALSE.
      BLADE  = .FALSE.
      IF(J.EQ.1) BOTTOM =.TRUE.
      IF(J.EQ.M) TOP =.TRUE.
      BLADE = TOP
      IF(.NOT.BOTTOM) DELPHI(J)=-E(J)*DELPHI(J+1)+F(J)
      OV=OVEREL
      IF(KKK.LT.NOTYET)OV=UNDERL
```

```
      IF(EMACH(I,J).GE.1.)OV=SUPREL*(1.-D(J)*(1.-CI))
      DELPSV(I,J)= DELPHI(J)*OV
      J=J-1
      GOTO856
857   CONTINUE
C     END
      J=0

C      GO TO NEXT RADIATING LINE

      IF(.NOT.EXIT)GOTO272
      TOT=0.




C     IF ALL LINES ARE COMPLETED, THEN
C     DO (UPDATE OF FIELD VARIABLES)

C     PROCEDURE (UPDATE OF FIELD VARIABLES)

      DO 871 I =2,IEXIT
      DO 871 J1=1,M
      PHI(I ,J1)= PHI(I ,J1)+DELPSV(I ,J1)
871   IF(J1.GE.4.AND.I.GT.1.AND.(ABS(DELPSV(I,J1)).LT.TEST*AVGFAC*GUESS
     1/FLOAT(IEXIT-1)/FLOAT(M-3).OR.KKK.LT.NOTYET))
     1TOT=TOT+ABS(DELPSV(I,J1))
      SSFAC=RADLE/FIXR*S*AQINF1*SIN(LAMDA0+BETA1)
      DO 880 I1=2,IEXIT
      IF(I1.EQ.NOZ) GOTO 880
C     IF(I1.NE.NOZ) THEN
      SFAC= SSFAC*FLOAT(ISIGN(1,I1-NOZ))

C      CORRECT THE DUMMY LINES BEYOND PERIODIC BOUNDARY USING PERIODICIT
C     Y
C      CONDITION.
      DO 878 J1=1,3
878   PHI(I1,J1)=PHI(IEXIT+2-I1,8-J1)+SFAC
C     END
880   CONTINUE

C       UPDATE DUMMY LINES ON EACH SIDE OF BRANCH C

      DO 890 J1=1,M
      PHI(1,J1)=
```

```
      1PHI(IEXIT-1,J1)-CIRCO
C     1PHI(IEXIT-1,J1)-RADLE/FIXR*S*(AQINF1*SIN(LAMDAO+BETA1)-AQINF2*SIN(
C     1LAMDAO+BETA2)/P1OP2)
890    PHI(IEXIT+1,J1)=PHI(3,J1)+
      1CIRCO
C     1          RADLE/FIXR*S*(AQINF1*SIN(LAMDAO+BETA1)-AQINF2*SIN(LAMDAO+
C     1BETA2)/P1OP2)
C      END


C      DO (CONVERGENCE TESTS)


C       PROCEDURE (CONVERGENCE TESTS)

       RELAXD=.FALSE.
       NOZO2=NOZ/2
       MO2=M/2
       AVGFAC=FLOAT(IEXIT-2)*FLOAT(M-3)*ABS(PHI(NOZO2,MO2))
       TEST=TOT/AVGFAC
       CIRCN=PHI(IEXIT,M)-PHI(2,M)
       DCIRC=(CIRCN-CIRCO)/CIRCOO
       IF(TOL.GT.0..AND.TOL.LT.1..AND.ABS(TEST).LT.TOL)RELAXD =.TRUE.
       IF(TOL.GT.1..AND.KK.GE.IFIX(TOL))RELAXD=.TRUE.
       TRUNC=ABS(TOL)
       IF(TOL.LT.0..AND.ABS(DCIRC).LT.TRUNC.AND.KKK.GT.50)RELAXD=.TRUE.
       IF(KKK.GE.KKMAX)RELAXD=.TRUE.
       TESMAC=0.
       DO 909 I=2,IEXIT
       RF=(.5*(DUMBX(I,M)+DUMBX(I-1,M)))**(1./2./EXPO)
       HIMAC=.5*(EMACH(I,M)+EMACH(I-1,M))*RF
       IF(HIMAC.GT.TESMAC.AND.ABS(X(I,M)/COS(LAMDAO))
      1.LT..44)TESMAC=HIMAC
909    CONTINUE
       IF(MOD(KK,IT).EQ.0.OR.BUG.OR.RELAXD)
      1WRITE(6,99) KK,TEST,TRANSP,TESMAC,CIRCN,DCIRC,OVEREL,SUPREL
99     FORMAT(1H ,4X,I4,4X,7E14.6)
C      END
       I=0


       IF(.NOT.RELAXD)GOTO269


C       OTHERWISE
C      DO (SURFACE FLOW CALCULATION FOR FINISHED GRID)
```

```
C       PROCEDURE (SURFACE FLOW CALCULATION FOR FINISHED GRID)

        WRITE(6,781) M3,ITIEX
        ICOUNT=0
        IDEX=1
        DO 937 I=1,IEXIT
        I1M=I-1
        IM1=I1M
        IF(IM1.EQ.0)IM1=2
        FIRSTJ=M-2
        LINEI=I


C       DO (VELOCITY CALCULATION ALONG RADIATING LINE)
        CALL SPEEDS(LINEI,FIRSTJ)


937     IF(I1M.GT.0)
       1WRITE(6,782)I1M,XOCX,FADBOP,X(I,M),Y(I,M),D(M),BOPR,
       1PHI(I,M),E(M),F(M),DELPHI(M)
781     FORMAT(1H0/1H0,22X,
       1'FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE ',I3,
       1' BY ',I3,' MESH '//'  RADIATING',50X,'  STATIC  '/'  LINE NO. ',
       1'    X/CX      Y/CX       X         Y          MACH    PRES ',
       1'COEF     PHI       XVEL       YVEL      DENSITY '/)
782     FORMAT(I6,4X,10F10.5)
C       END

        RETURN



        ENTRY KEEPER

C       DO (OUTPUT OF PHI AND CALCULATION OF FINAL VELOCITY FIELD)

C       PROCEDURE (OUTPUT OF PHI AND CALCULATION OF FINAL VELOCITY FIELD)

        REWIND 7
10      FORMAT(5E15.8)

        ENT=1.
```

```
      EXI=1.
      IF(ALLOUT)WRITE(6,131) M3,ITIEX
131   FORMAT(1H1,38X,' CALCULATED FLOW FIELD AT ALL GRID POINTS '/1H0,
     138X,
     1'SURFACE CONTOUR NO.  1  = PERIODIC BOUNDARY'/40X,'SURFACE CONT',
     1'OUR NO. ',
     1I3,' = BLADE SURFACE'/1H0,31X,' RADIATING LINE NO.  1 BEGINS AT',
     1' DOWNSTREAM',
     1' INFINITY '/42X,' AND IS REPEATED AS LINE NO. ',I3)
      IF(ALLOUT)WRITE(6,13)
13    FORMAT(1H0,15X,'SURFACE ',74X,'FLOW     STATIC'/16X'CONTOUR    ',
     1'X/CX        Y/CX     ',
     1'    PHI        XVEL        YVEL    DENSITY      MACH       ANGLE',
     1'   PRES COEF'/15X,' NUMBER')
      IDEX=1
      ICOUNT=0
      DO 969 I=1,IEXIT
      I1M=I-1
      IF(I.GT.1.AND.ALLOUT)WRITE(6,14) I1M
14    FORMAT('ORADIATING LINE'/'    NUMBER ',I3)
      IM1= I-1
      IF(IM1.EQ.0) IM1=2
      FIRSTJ=1
      LINEI=I


C     DO (VELOCITY CALCULATION ALONG RADIATING LINE)
      CALL SPEEDS(LINEI,FIRSTJ)


969   CONTINUE
      DO 972 I=2,IEXIT
972   E(I)=.5*(DELPSV(I,4)+DELPSV(IEXIT+2-I,4))
      DO 975 I=2,IEXIT
975   DELPSV(I,4)=E(I)
      REWIND 13
      MM3=M-3
      NNOZ=NOZ-1

C      WRITE THE PLOT DATA SAVE FILE

      WRITE(13,1)  NNOZ,MM3
1     FORMAT(2I5)
      WRITE(13,2)((X(I,J),I=2,NZZ),J=4,M),((Y(I,J),I=2,NZZ),J=4,M)
      WRITE(13,2)((DUMBX(I,J),I=2,NZZ),J=4,M),((DUMBY(I,J),I=2,NZZ),
     1J=4,M),((DELPSV(I,J),I=2,NZZ),J=4,M)
      WRITE(13,2) ((TENOLF(I,J),I=2,NZZ),J=4,M)
```

```
      IF(BUG3)
     1WRITE(6,2)((X(I,J),I=2,NZZ),J=4,M),((Y(I,J),I=2,NZZ),J=4,M)
      IF(BUG3)
     1WRITE(6,2)((DUMBX(I,J),I=2,NZZ),J=4,M),((DUMBY(I,J),I=2,NZZ),
     1J=4,M)
C     END

      RETURN







      END
      SUBROUTINE SPEEDS(I,MM2)

C      VELOCITY CALCULATION ALONG ONE RADIATING LINE (I), BEGINNING AT J
C     =
C      MM2 AND GOING TO J=M (THE BLADE SURFACE).    ARRAYS DELPHI, DELPS
C     V
C      D,E,F,PRIORS,DUMBY, AND DUMBX ARE REUSED (OVERWRITTEN) IN THE PRO
C     CESS.



      LOGICAL INLET,BLADE,EXIT,BOTTOM,TOP
      LOGICAL DOSURF,SMOOTH,FINEST,RESTAR,ALLOUT
      REAL LAB,LAD,MINF,LENGTH,JACOB,LAMDAO
      REAL MACH(8)
      DOUBLE PRECISION CAPK,CAPKP
      COMMON/PARAM/RESTAR,NOZIN,BETA1,BETA2,QINF1,CI,CII,GAMMA,MINF,EM
     1,DELTA,BETA,TOL,BUG,BUG2,BUG3,IT,GUESS,OVEREL,FIXR,NOWREL,NOTYET
     1,DAMP,SUPREL,S,WAKE,B2OB1,UNDERL,RADRAT,RADLE,OMEGA,FLOCO,VAXIAL,
     1ALLOUT
      COMMON/QUASI/ RADIUS(100,30),HEIGHT(100,30)
      COMMON/ENTIRE/  X(100,30),Y(100,30),PHI(100,30),DELPSV(100,30)
      COMMON/GEOM2/ZETA(100),ETA(100)
      COMMON/CROSS/ ZMACH(100,30)
      COMMON/GEOM/NOZ,M,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK,
     1CAPKP,PI
     1,RNK,VB,KN,NED,BB,DOSURF,RFAC,SOLID,MPLS
     1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THETT,THETL
     1,FINEST
```

```
      COMMON/CALVEL/FAKEFI(4,6),FAKFI(4,6),RHOCON,QLIM,EXPO,ROTATN,
     1ROTROT,IM1,IEXIT,PRIORS(14,30),INDEX(100,30),I1DEX(32),FAKEU(100),
     1FAKEV(100),ICOUNT,IDEX,AMY,AMZ
      COMMON/VELOUT/D(100),E(100),F(100),DELPHI(100),TENOLF(100,30),
     1DUMBX(100,30),DUMBY(100,30),BOPR,FADBOP,XOCX,KK
      DIMENSION RHOSS(8), FI1(8),FI2(3),FI3(3),FI9(8),SPAN(8),
     1EMACH(100,30)
     2,COSX(8),COSY(8),LENGTH(8),DEDX(8),DZDX(8),DEDY(8),
     3DZDY(8),T(8),DPHIDZ(8),DPHIDE(8),U(8),V(8),UN(8),RHO(8),
     4RHOUL(8),ONOFF(8),DELPM1(3)
     5,HITE(8)

      COMMON/HIBALL/ SUPERG(46,30,100)

      XDIF(A,B,C,D)=.5*(A+B-C-D)




C     PROCEDURE (VELOCITY CALCULATION ALONG RADIATING LINE)
C     DO (AXIAL LOCATION INDICATORS)
      INLET=.FALSE.
      EXIT  = .FALSE.
      IF(I.EQ.1) INLET =.TRUE.
      IF( I.EQ. IEXIT) EXIT=.TRUE.
      XLOW=0.
      XHI =0.
      YLOW=1.E10
      DO 1111 I1=1,IEXIT
      IF(XLOW.LT.X(I1,M))GOTO11
      XLOW=X(I1,M)
11    IF(XHI.GT.X(I1,M))GOTO111
      XHI=X(I1,M)
111   IF(YLOW.LT.Y(I1,M))GOTO1111
      YLOW=Y(I1,M)
1111  CONTINUE
      DO 1427 J=MM2,M
      J3M=J-3
      JM1=J-1
      IF(JM1.EQ.0) JM1=2
C     DO (VERTICAL LOCATION INDICATORS)
      BOTTOM = .FALSE.
      TOP    = .FALSE.
      BLADE  = .FALSE.
      IF(J.EQ.1) BOTTOM =.TRUE.
      IF(J.EQ.M) TOP =.TRUE.
      BLADE = TOP
```

```
C     DO (GEOMETRY ADVANCE)                                         QS018730
      SPAN(1)=XDIF(RADIUS(I,J),RADIUS(I+1,J),0.,0.)                 QS018740
      IF(I.EQ.IEXIT)SPAN(1)=RADIUS(I,J)                             QS018750
      SPAN(1)=SPAN(1)/FIXR                                          QS018760
      SPAN(2)=XDIF(RADIUS(I,J),RADIUS(I,J+1),0.,0.)                 QS018770
      SPAN(2)=SPAN(2)/FIXR                                          QS018780
      SPAN(3)=SPAN(2)                                               QS018790
      SPAN(8)=SPAN(1)                                               QS018800
      COSX(1)=SUPERG(1,J,I)                                         QS018810
      COSX(2)=SUPERG(2,J,I)                                         QS018820
      COSX(3)=SUPERG(3,J,I)                                         QS018830
      COSX(8)=SUPERG(4,J,I)                                         QS018840
      COSY(1)=SUPERG(5,J,I)                                         QS018850
      COSY(2)=SUPERG(6,J,I)                                         QS018860
      COSY(3)=SUPERG(7,J,I)                                         QS018870
      COSY(8)=SUPERG(8,J,I)                                         QS018880
      LENGTH(1)=SUPERG(9,J,I)                                       QS018890
      LENGTH(2)=SUPERG(10,J,I)                                      QS018900
      LENGTH(3)=SUPERG(11,J,I)                                      QS018910
      LENGTH(8)=SUPERG(12,J,I)                                      QS018920
      EXDIF1=SUPERG(13,J,I)                                         QS018930
      EXDIF2=SUPERG(14,J,I)                                         QS018940
      YDIF1=SUPERG(15,J,I)                                          QS018950
      YDIF2=SUPERG(16,J,I)                                          QS018960
      SLEN=SUPERG(17,J,I)                                           QS018970
      SLEN2=SUPERG(18,J,I)                                          QS018980
      ZETDF1=SUPERG(19,J,I)                                         QS018990
      ZETDF2=SUPERG(20,J,I)                                         QS019000
      ETADF1=SUPERG(21,J,I)                                         QS019010
      ETADF2=SUPERG(22,J,I)                                         QS019020
      DEDX(1)=SUPERG(23,J,I)                                        QS019030
      DEDX(2)=SUPERG(24,J,I)                                        QS019040
      DEDX(3)=SUPERG(25,J,I)                                        QS019050
      DEDX(8)=SUPERG(26,J,I)                                        QS019060
      DEDY(1)=SUPERG(27,J,I)                                        QS019070
      DEDY(2)=SUPERG(28,J,I)                                        QS019080
      DEDY(3)=SUPERG(29,J,I)                                        QS019090
      DEDY(8)=SUPERG(30,J,I)                                        QS019100
      DZDX(1)=SUPERG(31,J,I)                                        QS019110
      DZDX(2)=SUPERG(32,J,I)                                        QS019120
      DZDX(3)=SUPERG(33,J,I)                                        QS019130
      DZDX(8)=SUPERG(34,J,I)                                        QS019140
      DZDY(1)=SUPERG(35,J,I)                                        QS019150
      DZDY(2)=SUPERG(36,J,I)                                        QS019160
      DZDY(3)=SUPERG(37,J,I)                                        QS019170
      DZDY(8)=SUPERG(38,J,I)                                        QS019180
      T(1)=SUPERG(39,J,I)                                           QS019190
      T(2)=SUPERG(40,J,I)                                           QS019200
```

```
      T(3)=SUPERG(41,J,I)                                               QS019210
      T(8)=SUPERG(42,J,I)                                               QS019220
C     CELL   II,   POINT   C                                            QS019230
      T(5) =   PRIORS(1,J)*ONOFF(5)                                     QS019240
      T(4) =   PRIORS(2,J)*ONOFF(4)                                     QS019250
      PRIORS(1,J)  =  -  T(8)                                           QS019260
      PRIORS(2,J)  =  -  T(1)                                           QS019270
C     DO  (FIELD GRADIENTS)                                             QS019280
C     CELL II,   POINT C                                                QS019290
      DO  1214  I1=1,3                                                  QS019300
      FI1(I1)=PHI(I+1,J)                                                QS019310
      FI2(I1)=PHI(I+1,J+1)                                              QS019320
      FI3(I1)=PHI(I   ,J+1)                                             QS019330
 1214 FI9(I1)=PHI(I   ,J   )                                            QS019340
      FI4     =PHI(IM1,J+1)                                             QS019350
      FI5     =PHI(IM1,J   )                                            QS019360
      FI6     =PHI(IM1,JM1)                                             QS019370
      FI7     =PHI(I   ,JM1)                                            QS019380
      FI8     =PHI(I+1,JM1)                                             QS019390
      FI1(8)=PHI(I+1,J)                                                 QS019400
      FI9(8)=PHI(I   ,J)                                                QS019410
      I10=I-IEXIT+3                                                     QS019420
C     END                                                               QS019430
C     END                                                               QS019440
C     CELL I  ,  POINT  A                                               QS019450
 2501 DPHIDZ(1)=  XDIF(FI3(1),FI2(1),FI9(1),FI1(1))/ZETDF1              QS019460
      DPHIDE(1) =    (FI1(1)-FI9(1))/ETADF1                             QS019470
C     CELL  I,  POINT  B                                                QS019480
      DPHIDE(2)=XDIF(FI1(2),FI2(2),FI9(2),FI3(2))  /ETADF1              QS019490
      DPHIDZ(2)=(FI3(2)-FI9(2))/ZETDF1                                  QS019500
C     CELL  II,  POINT  B                                               QS019510
      DPHIDE(3)=XDIF(FI5,FI4,FI9(3),FI3(3))  /ETADF2                    QS019520
      DPHIDZ(3)=(FI3(3)-FI9(3))/ZETDF1                                  QS019530
C     CELL  IV,  POINT  A                                               QS019540
      DPHIDE(8) =    (FI1(8)-FI9(8))/ETADF1                             QS019550
      DPHIDZ(8) =XDIF(FI7,FI8,FI9(8),FI1(8))/ZETDF2                     QS019560
C     DO  (BOUNDARY CONDITION SWITCHES)                                 QS019570
      DO  1306  I1=1,8                                                  QS019580
 1306 ONOFF(I1)=1.                                                      QS019590
      IF(.NOT.INLET.AND..NOT.BOTTOM.AND..NOT.TOP  ) GOTO2060            QS019600
      IF(TOP)IS=1                                                       QS019610
      IF(INLET)IS=3                                                     QS019620
      IF(BOTTOM)IS=5                                                    QS019630
      IE=IS+3                                                           QS019640
      DO  2059  I1=IS,IE                                                QS019650
 2059 ONOFF(I1)=0.                                                      QS019660
C     DO  (VELOCITIES AND DENSITIES)                                    QS019670
 2060 U(6)  =  U(3)                                                     QS019680
```

```
      V(6) = V(3)                                                       QS019690
      U(7) = U(2)                                                       QS019700
      V(7) = V(2)                                                       QS019710
      DO 1337 I1 =1,8                                                   QS019720
      IF(I1.GE.4.AND.I1.NE.8) GOTO 1337                                 QS019730
C     IF(I1.LT.4.OR.I1.EQ.8) THEN                                       QS019740
      U(I1)=DPHIDE(I1)*DEDX(I1)+ DPHIDZ(I1)*DZDX(I1)                    QS019750
      V(I1)=DPHIDE(I1)*DEDY(I1)+ DPHIDZ(I1)*DZDY(I1)                    QS019760
      IF(INDEX(I,J).EQ.0)GOTO5411                                       QS019770
      ICOUNT=ICOUNT+1                                                   QS019780
      I1S=I1DEX(ICOUNT)                                                 QS019790
      IF(I1.LE.I1S)I1P=I1S                                              QS019800
      IDIF=INDEX(I,J)+I1S-1                                             QS019810
      IF(I1S.EQ.1.AND.I1P.EQ.IDIF)I1P=8                                 QS019820
      IF(I1.NE.I1P)GOTO5410                                             QS019830
      IF(I1P.GT.IDIF.AND.I1P.NE.8)GOTO5410                              QS019840
      V(I1)=FAKEV(IDEX)                                                 QS019850
      U(I1)=FAKEU(IDEX)                                                 QS019860
      IDEX=IDEX+1                                                       QS019870
      I1P=I1P+1                                                         QS019880
5410  IF(I1.NE.8)ICOUNT=ICOUNT-1                                        QS019890
5411  V(I1)=V(I1)-SPAN(I1)*FIXR/RADLE*ROTATN                            QS019900
      UN(I1)=U(I1)*COSX(I1)+V(I1)*COSY(I1)                              QS019910
C     END                                                               QS019920
1337  CONTINUE                                                          QS019930
      IF(.NOT.TOP) GOTO 2064                                            QS019940
C     IF(TOP) THEN                                                      QS019950
C     U(8)=(DPHIDE(8)*ETADF1-RADIUS(I,J)/RADLE*ROTATN*YDIF1)*EXDIF1/SLENQS019960
C     V(8)=(DPHIDE(8)*ETADF1-RADIUS(I,J)/RADLE*ROTATN*YDIF1)*YDIF1/SLEN QS019970
      ROTATX=ROTATN*RADIUS(I,J)/RADLE                                   QS019980
C     U8=SQRT((DPHIDE(8)*ETADF1/EXDIF1)**2+(DPHIDE(8)*ETADF1/YDIF1-     QS019990
C    1-ROTATX)**2)*EXDIF1/SQRT(SLEN)                                    QS020000
      U8=SQRT(U(8)*U(8)+V(8)*V(8))*ABS(EXDIF1)/SQRT(SLEN)               QS020010
      U(8)=SIGN(U8,U(8))                                                QS020020
      V(8)=U(8)*YDIF1/EXDIF1                                            QS020030
      UN(8)=U(8)*COSX(8)+V(8)*COSY(8)                                   QS020040
C     END                                                               QS020050
2064  U(4) =  PRIORS(3,J)                                               QS020060
      U(5) =  PRIORS(4,J)                                               QS020070
      V(4) =  PRIORS(5,J)                                               QS020080
      V(5) =  PRIORS(6,J)                                               QS020090
      PRIORS(3,J)=U(1)                                                  QS020100
      PRIORS(4,J)=U(8)                                                  QS020110
      PRIORS(5,J)=V(1)                                                  QS020120
      PRIORS(6,J)=V(8)                                                  QS020130
      RHO(7)=RHO(2)                                                     QS020140
      RHO(6)=RHO(3)                                                     QS020150
      RHO(4)=PRIORS(7,J)                                                QS020160
```

```
      RHO(5)=PRIORS(8,J)
      ONSUM=0.
      DO 1391 I1=1,8
C     IF(KK.LE.2.OR.MOD(KK,2).EQ.0) THEN
      IF(I1.GE.4.AND.I1.NE.8) GOTO 2066
C     IF(I1.LT.4.OR.I1.EQ.8) THEN
      RHOFAC=1.
      RHO(I1)=1.
      IF((I.EQ.2.OR.I.EQ.IEXIT).AND.J.LE.4.AND.ROTATN.NE.0.
     1.OR.KK.LE.NOTYET) GOTO 2066
      QSQAR=U(I1)*U(I1)+(V(I1)*V(I1))
      IF(QSQAR.GE.QLIM)QSQAR=.99*QLIM
      RHOFAC=RHOCON*(1.-QSQAR)*ONOFF(I1)+1.
      RHO(I1)=1.-AMZ*(QSQAR-1.)+AMY*(QSQAR-1.)*(QSQAR-1.)
      IF(QSQAR.LT..65.OR.QSQAR.GT.1.44)RHO(I1)=RHOFAC**EXPO
      ROTATX=SPAN(I1)*FIXR
      IF(ROTROT.NE.0..AND.J.GT.3)
     1RHO(I1)=(RHO(I1)**(1./EXPO)+RHOCON*ROTROT*((ROTATX/RADLE)
     1**2-1.)*ONOFF(I1))**EXPO
C     END
C     END
2066  QSQAR=U(I1)*U(I1)+V(I1)*V(I1)
      IF(I.GT.1)
     1MACH(I1)=MINF*SQRT(QSQAR)/RHO(I1)**(1./EXPO)
      IF(MACH(I1).GT.2.)MACH(I1)=2.
1391  CONTINUE
      PRIORS(7,J)=RHO(1)
      PRIORS(8,J)=RHO(8)
      IF(I.LE.1.OR.J.LE.3) GOTO 1427
C     IF(I.GT.1.AND.J.GT.3) THEN
      F(J)=1.
      F(J)=0.
      ONSUM=ONOFF(1)+ONOFF(8)+ONOFF(4)+ONOFF(5)
      E(J)=(U(1)*ONOFF(1)+U(8)*ONOFF(8)+U(4)*ONOFF(4)+U(5)
     1*ONOFF(5))/ONSUM
      F(J)=(V(1)*ONOFF(1)+V(8)*ONOFF(8)+V(4)*ONOFF(4)+V(5)
     1*ONOFF(5))/ONSUM
      DUMBX(I,J)=E(J)
      DUMBY(I,J)=F(J)
      EFFAC=1.-E(J)*E(J)-F(J)*F(J)
      DELPHI(J)=(RHO(1)*ONOFF(1)+RHO(8)*ONOFF(8)+RHO(4)*ONOFF(4)+RHO(5)
     1*ONOFF(5))/ONSUM
      D(J)=0.
      D(J)=MINF*SQRT((F(J)*F(J)+E(J)*E(J))/DELPHI(J)**(1./EXPO))
      IF(J.LE.6.AND.(I.LE.4.OR.I.GE.IEXIT-2)) D(J)=EM
      IF(J.LE.6.AND.I.GE.NOZ-1.AND.I.LE.NOZ+1)D(J)=MINF
      DELPSV(I,J)=D(J)
C     END
```

```
2076  IF(E(J).NE.0.)BOPS=ATAN2(F(J),E(J))*180./PI
      BOPR=-999.
      IF(D(J).NE.0.)
     1BOPR=2./GAMMA*(DELPHI(J)*(1.-EFFAC)/D(J)/D(J)-1./MINF/MINF)
      IF(MINF.LT..25)BOPR=1.+DELPHI(J)*(EFFAC-1.)
      XOCX=(X(I,J)-XLOW)/(XHI-XLOW)
      FADBOP=(Y(I,J)-YLOW)/(XHI-XLOW)
      IF(ALLOUT.AND.MM2.EQ.1)
     1WRITE(6,12) J3M,XOCX,FADBOP,PHI(I,J),E(J),F(J),DELPHI(J)
     1,D(J),BOPS,BOPR
      TENOLF(I,J)=BOPR
C     END
1427  CONTINUE
12    FORMAT(15X,I4,4X,9F10.5)
      RETURN
      END
      SUBROUTINE WRAPUP(ZETA,ETA,S)


C             GEOMETRY GENERATION FOR CASCADE POTENTIAL FLOW SOLVER



      LOGICAL DOSURF,SMOOTH,BUG2,ORTHO
      REAL LAMDAO
      COMPLEX H1,H2,ORIGIN,CDWISE
      COMPLEX AI/(0.,1.)/
      DOUBLE PRECISION CAPK,CAPKP
      DIMENSION ZETA(100),ETA(100),DUMBE(100),H1(100),H2(100)
      DIMENSION XB(200),YB(200),DUMBX(100,30)
      COMMON/ENTIRE/  X(100,30),Y(100,30),PHI(100,30),DUMBY(100,30)
      COMMON/GEOM/NOZ,M,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK
     1,CAPKP,PI
     1,RNK,VB,KN,NED,BB,DOSURF,RFAC,SOLID,MPLS
     1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THETT,THETL
      NAMELIST/INSTUF/H1,H2,IMAXY,BUG2
      DATA ORTHO/.FALSE./,BUG2/.FALSE./

      MAGIC=0
      MAGICM=0


      REWIND 18

C      RECORD OLD COMPUTATIONAL VARIABLES FOR INTERPOLATION IN FILPHI

      WRITE(18) ETA,ZETA
      PI=3.14159265
```

```
      SO=CHORD
C      IF(NOZ.LT.0) OBTAIN MESH FROM STORAGE FILE

      IF(NOZ.LE.0) GOTO201
C     IF(NOZ.GT.0) GENERATE A NEW MESH
      IEXIT=2*NOZ-1
      NZZ=IEXIT
      DOSURF=.TRUE.
      IF(NED.LE.0)DOSURF=.FALSE.
      NED=IABS(NED)
      IF(DOSURF)READ(5,INSTUF,END=15)
C     WRITE(6,12)
12    FORMAT(37HTHE FOLLOWING IS THE FIRST ECHO PRINT)
C     WRITE(6,INSTUF)
15    IF(DOSURF)READ(5,INSTUF,END=16)
      WRITE(6,11)
11    FORMAT(38HTHE FOLLOWING IS THE SECOND ECHO PRINT)
      WRITE(6,INSTUF)
16    IF(KN.EQ.0)ORTHO=.TRUE.
      IF(ORTHO) GOTO203
C     IF(.NOT.ORTHO) REARRANGE H2 INPUT INTO H1(UPPER SURFACE), AND H2(L
C     OWER SURFACE)
C      UNSTAGGERED DATA TABLES, FOR THE INTERPOLATION SCHEME

      NEDKN1=NED-KN+1
      ORIGIN=H2(NEDKN1)
      DO 205 I1=1,KN
      I5=NED-KN+I1
      H1(I1)=H2(I5)
      CDWISE =(H1(I1)-ORIGIN)*CEXP(-AI*LAMDA0)
      H1(I1)=CDWISE
205   CONTINUE
      DO 206 I1=1,NEDKN1
      I6=NED-KN-I1+2
      IF(I6.LE.0)I6=1
206   H1(KN+I1)=H2(I6)
      DO 209 I1=1,NEDKN1
      H2(I1)=H1(KN+I1)
      CDWISE=(H2(I1)-ORIGIN)*CEXP(-AI*LAMDA0)
      H2(I1)=CDWISE
209   CONTINUE
      NED=NED+1
      GOTO233

C      IF(ORTHO) REARRANGE H2 INPUT TO START AT MINIMUM Y POINT RATHER T
C     HAN MAX X,
```

```
C       FOR ELECTROSTATIC ANALOG SCHEME

203     T2=1.E10
        IT2=0
        DO 2333 I1=1,NED
        IF(AIMAG(H2(I1)).GE.T2) GOTO2333
        IT2=I1
        T2=AIMAG(H2(IT2))
2333    CONTINUE
        DO 23333 I1=1,NED
        I7=IT2+I1-1
        IF(I7.EQ.NED)IT3=I1
        IF(I7.GT.NED)I7=I1-IT3+1
23333   H1(I1)=H2(I7)
        T2=-1.E10
        I2T=0
        DO 2033 I1=1,NED
        IF(AIMAG(H1(I1)).LE.T2)GOTO2033
        T2=AIMAG(H1(I1))
        I2T=I1
2033    H2(I1)=H1(I1)


        IMAXY=I2T

C       LOAD COMPLEX INPUT INTO XBOD AND YBOD ARRAYS

233     CONTINUE
        DO 213 K=1,NED
        KMKN=K-KN
        IF(K.GT.KN) GOTO217
C       IF(K.LE.KN) THEN
        XB(K)=REAL(H1(K))
        YB(K)=AIMAG(H1(K))
        GOTO213
217     XB(K)=REAL(H2(KMKN))
        YB(K)=AIMAG(H2(KMKN))
213     CONTINUE
        IF(CHORD.EQ.1.) GOTO219
C       IF(CHORD.NE.1.) NORMALIZE COORDINATES TO AERODYNAMIC CHORD
        DO 221 I2=1,NED
        XB(I2)=XB(I2)/CHORD
221     YB(I2)=YB(I2)/CHORD
        RLE=RLE/CHORD
        RTE=RTE/CHORD
        S=S/CHORD
        CHORD=1.
C       END
```

```
219   IF(.NOT.ORTHO) GOTO225
C     IF(ORTHO) DO ELECTROSTATIC ANALOG GRID

      DO 227 I2=1,NED
227   H1(I2)=CMPLX(XB(I2),YB(I2))
      WRITE(6,9)
9     FORMAT(1H1,40X,' ELECTROSTATIC ANALOG GRID GENERATED')
      CALL PJGRID(ZETA,ETA,H1,IMAXY,S)
      GOTO226

C      OTHERWISE DO INTERPOLATION SCHEME GRID

225   WRITE(6,10)
10    FORMAT(1H1,40X,' INTERPOLATION SCHEME GRID GENERATED')
      CALL GRIDDL(ZETA,ETA,XB,YB,S)

C      WRITE MESH POINTS STORAGE FILE

226   WRITE(13,1) NOZ,M,KN,NED
      WRITE(13,2)((X(I,J),I=1,NZZ),J=1,M),((Y(I,J),I=1,NZZ),J=1,M)
     1,(ETA(I),I=1,NZZ),(ZETA(J),J=1,M)


      GOTO202

C      FOR EXISTING MESH FILES, NORMALIZE PARAMETERS AND READ FILE

201   IF(CHORD.EQ.1.) GOTO231
C     IF(CHORD.NE.1.) THEN
      RLE=RLE/CHORD
      RTE=RTE/CHORD
      S=S/CHORD
      CHORD=1.
C     END
231   READ(23,1) NOZZ,MM,KN,NED
      MAGIC=IABS((NOZZ-1)/(NOZ+1))
      MAGICM=(MM-1)/(M-1)
      M=MM
      NOZ=NOZZ
      IEXIT=2*NOZ-1
      NZZ=IEXIT
      READ(23,2)((X(I,J),I=1,NZZ),J=1,M),((Y(I,J),I=1,NZZ),J=1,M)
```

```
     1,(ETA(I),I=1,NZZ),(ZETA(J),J=1,M)

C      IF FINEST MESH IS DESIRED, SKIP SELECTION PROCESS

      IF(MAGIC.LE.1.AND.MAGICM.LE.1) GOTO0202
C     IF(MAGIC.GT.1.OR.MAGICM.GT.1) SELECT AND LOAD GRID LINES DESIRED T
C     O FORM
C      A COARSE MESH
      DO 235 J=1,M,MAGICM
      JON=(J-1)/MAGICM+1
      ZETA(JON)=ZETA(J)
      DO 239 I=1,NZZ,MAGIC
      ION=(I-1)/MAGIC+1
      DUMBX(ION,JON)=X(I,J)
  239 DUMBY(ION,JON)=Y(I,J)
      IF(MOD(ION,2).EQ.0) ION=ION-1

C      RADIATING LINE TO UPSTREAM INFINITY MUST BE INCLUDED
      DUMBX(ION,JON)=X(NZZ,J)
      DUMBY(ION,JON)=Y(NZZ,J)

      NOZZ=(ION+1)/2

C      RADIATING LINE TO DOWNSTREAM INFINITY MUST BE INCLUDED
      DUMBX(NOZZ,JON)=X(NOZ,J)
  235 DUMBY(NOZZ,JON)=Y(NOZ,J)

      DO 243 I=1,ION
      IOLD=(I-1)*MAGIC+1
      ETA(I)=ETA(IOLD)

C      BLADE SURFACE MUST BE LAST CLOSED CURVE
      DUMBX(I,JON)=X(IOLD,M)
      DUMBY(I,JON)=Y(IOLD,M)

C      PERIODIC BOUNDARY MUST BE FIRST CLOSED CURVE
      DUMBX(I,1)=X(IOLD,1)
  243 DUMBY(I,1)=Y(IOLD,1)

      DUMBX(ION,JON)=X(NZZ,M)
      DUMBY(ION,JON)=Y(NZZ,M)
      DUMBX(ION,1)=X(NZZ,1)
      DUMBY(ION,1)=Y(NZZ,1)
      DUMBX(NOZZ,M)=X(NOZ,M)
      DUMBY(NOZZ,M)=Y(NOZ,M)
      DUMBY(NOZZ,1)=Y(NOZ,1)
      DUMBX(NOZZ,1)=X(NOZ,1)
```

```
      DO 251 J=1,JON
      DO 251 I=1,ION
      X(I,J)=DUMBX(I,J)
251   Y(I,J)=DUMBY(I,J)
      ZETA(JON)=ZETA(M)
      ETA(ION)=ETA(NZZ)
      ETA(NOZZ)=ETA(NOZ)
      NZZ=ION
      IEXIT=NZZ
      NOZ=NOZZ
      M=JON
C      END OF SELECTION PROCESS
202   CONTINUE
      IF(.NOT.BUG2) GOTO255
C     IF(BUG2) THEN
      WRITE( 6,3) NOZ,M,KN,NED
3     FORMAT('0NOZ= ',I3,' M= ',I3,' KN= ',I3,' NED= ',I3)
      NEDM1=NED-1

C      FOR A NEW GEOMETRY GENERATION, CALCULATE THE SECOND DERIVATIVES O
C     N BLADE
C      SURFACE USING SMOOTHED INPUT COORDINATES

      DO 257 I=2,NEDM1
      DUMBX(I,15)=1.E06
      IF(XB(I).NE.XB(I-1).AND.XB(I).NE.XB(I+1).AND.
     1XB(I+1).NE.XB(I-1))
     1DUMBX(I,15)=((YB(I+1)-YB(I))/(XB(I+1)-XB(I))-(YB(I)-YB(I-1))/
     1(XB(I)-XB(I-1)))/(XB(I+1)-XB(I-1))*2.
257   CONTINUE

      WRITE(6,5)
      WRITE(6,55)
5     FORMAT(1H0,35X,'SMOOTHED BLADE SURFACE INPUT, HORIZONTAL CHORD')
55    FORMAT(1H0,35X,'   I          X              Y          2ND DERIV.    '/)
6     FORMAT(1H0,20X,'X GRID POINTS, AS STORED ON MESH FILE'/)
7     FORMAT(1H0,20X,'Y GRID POINTS, AS STORED ON MESH FILE'/)
4     FORMAT(35X,I4,3E13.5)
      WRITE(6,4) (I,XB(I),YB(I),DUMBX(I,15),I=1,NED)
      WRITE(6,6)
      WRITE(6,2)((X(I,J),I=1,NZZ),J=1,M)
      WRITE(6,7)
```

```
      WRITE(6,2) ((Y(I,J),I=1,NZZ),J=1,M)                               QS023530
1     FORMAT(4I5)                                                       QS023540
14    FORMAT(1H ,14E9.3)                                                QS023550
2     FORMAT(6E13.5)                                                    QS023560
                                                                        QS023570
                                                                        QS023580
                                                                        QS023590
255   IF(KN.NE.0)GOTO 261                                               QS023600
C     IF(ORTHO) TRANSLATE THE ORIGIN OF X POINTS TO MIDCHORD (INTERP SCHQS023610
C     EME                                                               QS023620
C      ALREADY TRANSLATED                                               QS023630
                                                                        QS023640
      DO 263 I1=1,NZZ                                                   QS023650
      DO 263 J1=1,M                                                     QS023660
263   X(I1,J1)=X(I1,J1)-.5                                              QS023670
                                                                        QS023680
C      REARRANGE (X,Y) TABLES SO THAT X(2,J),Y(2,J) AND X(NZZ+1,J),Y(NZZQS023690
C     +1,J) GO                                                          QS023700
C      TO DOWNSTREAM INFINITY, AND X(NOZ+1,J),Y(NOZ+1,J) GOES TO UPSTREAQS023710
C     M INFINITY                                                        QS023720
C      NUMBERING THE LINES CLOCKWISE AROUND THE BLADE.  AT PRESENT, LINEQS023730
C      1 GOES                                                           QS023740
C      TO UPS. INF.,  NOZ GOES TO DOWNS. INF. AND NZZ TO UPS. INF.      QS023750
C                                                                       QS023760
C      ALSO REALIGN ETA SO THAT VALUES WILL GO FROM ZERO (LINE 2) TO 2*CQS023770
C     APKP (LINE                                                        QS023780
C      NZZ+1), INSTEAD OF -CAPKP (LINE 1)  TO +CAPKP (LINE NZZ).        QS023790
                                                                        QS023800
261   DO 271 I1=1,NOZ                                                   QS023810
      DO 271 J1=1,M                                                     QS023820
      I2=I1+NOZ-1                                                       QS023830
      DUMBE(I1)=-ETA(I2)                                                QS023840
      DUMBX(I1,J1)=X(I1,J1)                                             QS023850
      DUMBY(I1,J1)=Y(I1,J1)                                             QS023860
      X(I1,J1)=X(I1+NOZ-1,J1)                                           QS023870
271   Y(I1,J1)=Y(I1+NOZ-1,J1)                                           QS023880
      NOZP1=NOZ+1                                                       QS023890
      DO 279 I1=NOZP1,NZZ                                               QS023900
      DO 279 J1=1,M                                                     QS023910
      I2=I1-NOZ+1                                                       QS023920
      DUMBE(I1)=-ETA(I2)+2.*ETA(1)                                      QS023930
      DUMBX(I1,J1)=X(I1,J1)                                             QS023940
      DUMBY(I1,J1)=Y(I1,J1)                                             QS023950
      X(I1,J1)=DUMBX(I1-NOZ+1,J1)                                       QS023960
279   Y(I1,J1)=DUMBY(I1-NOZ+1,J1)                                       QS023970
                                                                        QS023980
C      MOVE ALL VALUES UP 3 LOCATIONS IN J TO MAKE ROOM FOR 3 DUMMY ELLIQS023990
C     PSES                                                              QS024000
```

```
C        BEYOND THE PERIODIC LINE.                                         QS024010
                                                                           QS024020
         I1=IEXIT                                                          QS024030
2088     IF(I1.LT.1)GOTO2872                                               QS024040
2955     J1=M                                                              QS024050
2087     IF(J1.LT.1)GOTO2871                                               QS024060
         X(I1,J1+3)= X(I1,J1)                                              QS024070
         IF(I1.EQ.1)ZETA(J1+3)=ZETA(J1)                                    QS024080
287      Y(I1,J1+3)= Y(I1,J1)                                              QS024090
         J1=J1-1                                                           QS024100
         GOTO2087                                                          QS024110
2871     I1=I1-1                                                           QS024120
         GOTO2088                                                          QS024130
                                                                           QS024140
                                                                           QS024150
C        FINISH LOADING ETA; CREATE 3 DUMMY LINES USING PERIODIC RELATION QS024160
C        OF X,Y VALUES                                                     QS024170
                                                                           QS024180
2872     I1=IEXIT                                                          QS024190
295      IF(I1.LT.2)GOTO2951                                               QS024200
         ETA(I1)=DUMBE(I1)                                                 QS024210
         DO 299 J1=1,3                                                     QS024220
         SIGEN=1.                                                          QS024230
         IF(I1.LT.NOZ)SIGEN=-1.                                            QS024240
         IF(I1.EQ.NOZ)SIGEN=0.                                             QS024250
         X(I1,J1)=X( IEXIT-I1+1  ,8-J1)+S*SIN(LAMDAO) *SIGEN               QS024260
         IF(I1.EQ.2) ZETA(J1)=-ZETA(8-J1)                                  QS024270
299      Y(I1,J1)=Y( IEXIT-I1+1  ,8-J1)+S*COS(LAMDAO)*SIGEN                QS024280
         ETA(I1+1)=ETA(I1)                                                 QS024290
                                                                           QS024300
                                                                           QS024310
C        LOAD SPECIAL POINTS (J=4 NOW EQUALS PERIODIC LINE POINTS)         QS024320
                                                                           QS024330
         I1=I1-1                                                           QS024340
         GOTO 295                                                          QS024350
2951     ETA(2)=0.                                                         QS024360
         ETA(IEXIT+1)=2.*ETA(1)                                            QS024370
         X(IEXIT,1)=2.*X(IEXIT,4)                                          QS024380
         X(IEXIT,2)=1.5*X(IEXIT,4)                                         QS024390
         X(IEXIT,3)=1.2*X(IEXIT,4)                                         QS024400
         Y(IEXIT,1)=2.*Y(IEXIT,4)                                          QS024410
         Y(IEXIT,2)=1.5*Y(IEXIT,4)                                         QS024420
         Y(IEXIT,3)=1.2*Y(IEXIT,4)                                         QS024430
         X(NOZ,1)=2.*X(NOZ,4)                                              QS024440
         X(NOZ,2)=1.5*X(NOZ,4)                                             QS024450
         X(NOZ,3)=1.2*X(NOZ,4)                                             QS024460
         Y(NOZ,1)=2.*Y(NOZ,4)                                              QS024470
         Y(NOZ,2)=1.5*Y(NOZ,4)                                             QS024480
```

```
      Y(NOZ,3)=1.2*Y(NOZ,4)
      MP3=M+3

C      MOVE ALL DATA UP ONE I LOCATION TO MAKE ROOM FOR A REPEATED RADIA
C     TING LINE
C      NEXT TO DOWNSTREAM INFINITY LINE .

      DO 2103 J1=1,MP3
      I1=IEXIT
 2107 IF(I1.LT.1)GOTO21071
      X(I1+1, J1)=X(I1,J1)
      Y(I1+1, J1)=Y(I1,J1)
      DUMBX(I1,J1)=0.
      DUMBY(I1,J1)=0.
      I1=I1-1
      GOTO2107
21071 X(2,J1)=X(IEXIT+1,J1)
      Y(2,J1)=Y(IEXIT+1,J1)
      X(1,J1)=X(IEXIT   ,J1)
      Y(1,J1)=Y(IEXIT   ,J1)
      X(IEXIT+2,J1)=X(3,J1)
 2103 Y(IEXIT+2,J1)=Y(3,J1)


      ETA(IEXIT+2)=ETA(3)+2.*ETA(1)
      ETA(1)=-ETA(3)

C      UPDATE COUNTERS

      NZZ=NZZ+1
      IEXIT=IEXIT+1
      NOZ=NOZ+1
      M=M+3


      ZETA(M+1)=ZETA(M)+ZETA(M)-ZETA(M-1)
      I67=M-14+1
      I67=MAXO(I67,1)
      I671=I67+1



      IF(.NOT.BUG2) RETURN

C      FOR NEW GEOMETRY GENERATIONS, CALCULATE SECOND DERIVATIVES ON BLA
C     DE SURFACE
C      USING THE FINAL GRID POINTS.
```

```
      NZZM1=NZZ-1
      DO 2113 I=2,NZZM1
      DUMBX(I,15)=1.E06
      IF(X(I,M).NE.X(I-1,M).AND.X(I,M).NE.X(I+1,M).AND.
     1X(I+1,M).NE.X(I-1,M))DUMBX(I,15)=
     1((Y(I+1,M)-Y(I,M))/(X(I+1,M)-X(I,M))-(Y(I,M)-Y(I-1,M))/
     1(X(I,M)-X(I-1,M)))/(X(I+1,M)-X(I-1,M))*2.
2113  CONTINUE


      WRITE(6,8)
8     FORMAT(1H0,30X,' GRID OUTPUT POINTS ON BLADE SURFACE, ',
     1'HORIZONTAL CHORD'//36X'   I        X              Y',
     1'         2ND DERIV.    '/)
      DO 2119 I=2,NZZM1
      I1M=I-1
2119  WRITE(6,4) I1M,X(I,M),Y(I,M),DUMBX(I,15)
      BUG2=.FALSE.




      RETURN
      END
      SUBROUTINE GRIDDL(ET,ZET,XBOD,YBOD,S)


C                      INTERPOLATION SCHEME GRID GENERATOR



C       IF(STAG.NE.0) DO (INTERMEDIATE TRANSFORM SETUP, ZERO STAG, S=2PI)
C       DO (CALCULATION OF GEOMETRY CONSTANTS)
C       DO (KNOWN GRID COORDINATES, ET, ZET)
C            IF(SLP2.LT.0) THEN
C                 DO (FLATPLATE-ZERO STAGGER TRANSFORM, UPPER SURFACE)
C                 DO (FLATPLATE-ZERO STAGGER TRANSFORM, LOWER SURFACE)
C                 CALL TOSTAG(S,ZET,CHORD)
C            DO (CONCENTRATION OF RADIATING LINES, ZET)
C            DO (CONCENTRATION OF SURFACE CONTOURS, ET)
C       DO (FLATPLATE-ZERO STAGGER TRANSFORM, UPPER SURFACE)
C       DO (FLATPLATE-ZERO STAGGER TRANSFORM, LOWER SURFACE)
C       DO (SPECIAL POINTS)
C       DO (CIRCLE TRANSFM. FROM ZERO STAGGER TO INPUT VALUES)
C       DO (RECALCULATION OF GEOMETRY CONSTANTS)
C       DO (TRANSLATE ORIGIN OF INPUT BODY POINTS TO MIDCHORD)
C       DO (TRAILING EDGE FIX OF EXTRA CAMBER POINTS)
```

```
C       DO (ADDITION OF BLADE SURFACE ON CLEAN FLATPLATE MESH)
C       RETURN


        DIMENSION  ET(100),ZET(100),XBOD(200)    ,YBOD(200)
        REAL  K,LNK,M1,LAM,LAMDA0,M
        INTEGER ZNO,ENO,ZN2M1,ZN2,END,ENOPLS,CAMBER,ZNP1,ENOP1
        DOUBLE PRECISION MMDELK
        DOUBLE PRECISION  CAPK,CAPKP,DM,DM1
        LOGICAL DOSURF,SMOOTH
        COMMON/ENTIRE/  X(100,30),Y(100,30)
        COMMON/GEOM/ZNO,ENO,RLE,RTE,A,CHORB,XUPS,XDNS,STAG,CC,CAPK
       1,CAPKP,PI
       1,LNK,M1,KM,END,M,DOSURF,R,SOLID,ENOPLS
       1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,DUMBO(5),CAMBER
        XUPS=-1.E+01
        XDNS=1.E+01
        PI=3.14159265
        DELTA=0.
        ZNP1=ZNO+1
        LAMDA0=STAG
555     FORMAT(1H ,2E12.5)
        ENOPLS=ENO+ENO/CAMBER
        CHORD=CHORB-RLE/2.-RTE/2.
        XDNS= 25.*CHORD
        XUPS=-25.*CHORD
        SOLID=CHORB/S
        SOLFLT=CHORD/S
        SOLID0=SOLFLT
        IF(STAG.EQ.0.)GOTO3002

C       DO (INTERMECIATE TRANSFORM SETUP, I.E. ZERO STAG, S=2PI)

C       PROCEDURE (INTERMEDIATE TRANSFORM SETUP, I.E. ZERO STAG, S=2PI)
        ASTAG=ABS(STAG)
        CALL ALTERP(SOLFLT,ASTAG,ALFA,R)
        CHORD=ALOG((R*R+2.*R+1.)/(R*R-2.*R+1.))*2.
        SOLID0=CHORD/PI/2.
        S=2.*PI
        LAMDA0=0.
C       END

3002    CONTINUE
```

```
C      DO (CALCULATION OF GEOMETRY CONSTANTS)

C      PROCEDURE (CALCULATION OF GEOMETRY CONSTANTS)
       A= COS(LAMDAO)/2./SOLIDO
       K= EXP(-PI/2./A)
       LNK = -ALOG(K)
       M=K*K
       DM=M
       DM1=1.DO-DM
       M1=DM1
       CAPK=MMDELK(1,DM,IER)
       CAPKP=MMDELK(1,DM1,IER)
C      END



C      DO (KNOWN GRID COORDINATES, ET,ZET)

C      PROCEDURE (KNOWN GRID COORDINATES, ET,ZET)
       DELZ=CAPKP/(ZNO-1)
       DELE=CAPK*STABAC/(ENO-1)
       ZN2M1=2*ZNO-1
       ZET(1)=+CAPKP
       DO 217  NZ=2,ZN2M1
217    ZET(NZ)=ZET(NZ-1)-DELZ
       ET(1)=0.
       ZN2 =  2*ZNO
       DO 221  NE=2,ENOPLS
221    ET(NE)=ET(NE-1)+DELE
       ET(ENO)=STABAC*CAPK
       ZET(ZNO)=0.
       IF(SLP1.GE.0.)GOTO205


C      IF(SLP1.LT.0.) DO (CONCENTRATION OF GRID LINES IN KEY AREAS)
C      PROCEDURE (CONCENTRATION OF GRID LINES IN KEY AREAS)
       XTES=-1.E19
       ITS=1
       SLP1=-SLP1


C       IF SLP2.LT.0 SETUP THE PROPER STAGGERED FLATPLATE MESH TO FIND WH
C      ICH LINE
C       INTERSECTS T. E.


       IF(SLP2.GE.0..OR.S.NE.2.*PI) GOTO226
       DELTA=ABS(SLP2)
```

```
C       DO (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y UPPER SURFACE)
C       PROCEDURE (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y UPPER SURFACE)
        DO 2060 I=1,ZNO
        DO 2060 J=1,ENOPLS
        IF((I.EQ.1.OR.I.EQ.ZNO).AND.J.EQ.1) GOTO2060
        ETER=ET(J)

C       DO (YOUR BASIC TRANSFORMATION )
C       PROCEDURE (YOUR BASIC TRANSFORMATION )
        ZED=ZET(I)
        IF(ETER.GT.CAPK)ZED=-ZED
        IF(ETER.GT.CAPK) ETER=2.*CAPK-ETER
        IF(I.EQ.ZNO)ZED=0.
        ALLFNC=FUDS(ETER, ZED,M1)
        LAM   = BADZAC(M1)
        TH    = OWOW(M1)
        X(I,J)= CHORD*A/PI*(LNK-2.*ALOG(ALLFNC/LAM))
        IF(J.GT.ENO.AND.I.EQ.ZN2M1)X(I,J)=-CHORD-X(I,J)
        IF(J.GT.ENO.AND.I.EQ.ZNO)X(I,J)=CHORD-X(I,J)
        Y(I,J)= 2.*A*CHORD/PI*TH
        IF(X(I,J).GT.XDNS)X(I,J)=XDNS
        IF(X(I,J).LT.XUPS)X(I,J)=XUPS
C       END

2060    CONTINUE
C       END

C       DO (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y LOWER SURFACE)
C       PROCEDURE (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y LOWER SURFACE)
        DO 2071 I=ZNP1,ZN2M1
        IDIF=I-ZNO
        X(I,1)=X(ZNO-IDIF,1)-S*SIN(LAMDAO)
        Y(I,1)= -A*CHORD
        DO 2071 J=1,ENOPLS
        IF(I.EQ.ZN2M1.AND.J.EQ.1) GOTO2071
        ETER=ET(J)

C       DO (YOUR BASIC TRANSFORMATION )
C       PROCEDURE (YOUR BASIC TRANSFORMATION )
        ZED=ZET(I)
        IF(ETER.GT.CAPK)ZED=-ZED
        IF(ETER.GT.CAPK) ETER=2.*CAPK-ETER
        IF(I.EQ.ZNO)ZED=0.
        ALLFNC=FUDS(ETER, ZED,M1)
        LAM   = BADZAC(M1)
        TH    = OWOW(M1)
        X(I,J)= CHORD*A/PI*(LNK-2.*ALOG(ALLFNC/LAM))
```

```
      IF(J.GT.ENO.AND.I.EQ.ZN2M1)X(I,J)=-CHORD-X(I,J)
      IF(J.GT.ENO.AND.I.EQ.ZNO)X(I,J)=CHORD-X(I,J)
      Y(I,J)= 2.*A*CHORD/PI*TH
      IF(X(I,J).GT.XDNS)X(I,J)=XDNS
      IF(X(I,J).LT.XUPS)X(I,J)=XUPS
C     END

2071  CONTINUE
C     END

      CALL TOSTAG(S,ZET,CHORD)
      DO 234 I3=1,ZN2M1
      IF(X(I3,ENO).LE.XTES) GOTO234
C     WRITE(6,4445) XTES,I3,X(I3,ENO)
4445  FORMAT(E13.5,I6,E13.5)
      XTES=X(I3,ENO)
      ITS=I3
234   CONTINUE
      EF=ZET(ITS)/CAPKP
      IF(SLP1.EQ.10.) EF=1.-EF
      SLP1=(1.-DELTA/EF/EF*(.6667-.2/EF/EF))/
     1(1.-1./EF/EF*(.6667-.2/EF/EF))
      SLP2=3.-SLP1*(2.-1./2.5/EF**4)-DELTA/2.5/EF**4
226   CONTINUE

C      CONCENTRATE THE RADIATING LINES  (COMPUTATIONAL COORD ZET)

C     DO (CONCENTRATION POLYNOMIAL)
C     PROCEDURE (CONCENTRATION POLYNOMIAL)
      CON1=2.5*(1.-SLP1)/CAPKP**2
      CON2=(SLP2-SLP1)/(2.*CAPKP**2)
      CON1=CON1-CON2
      CON3=(SLP2-SLP1)/(2.*CAPKP**4)
      CON4=1.5*(1.-SLP1)/(CAPKP**4)
      CON3=CON3-CON4
C     END

      DO 237 I3=1,ZN2M1
      CAPFAC=0.
      ZETER1=ZET(I3)
237   ZET(I3)=SLP1*ZETER1+CON1*ZETER1**3+CON3*ZETER1**5-CAPFAC
      IF(DELTA.EQ.0.) GOTO241
      EF=1.-EF
      SLP1=(1.-DELTA/EF/EF*(.6667-.2/EF/EF))/
     1(1.-1./EF/EF*(.6667-.2/EF/EF))
      SLP2=3.-SLP1*(2.-1./2.5/EF**4)-DELTA/2.5/EF**4
```

```
C       DO (CONCENTRATION POLYNOMIAL)
C       PROCEDURE (CONCENTRATION POLYNOMIAL)
        CON1=2.5*(1.-SLP1)/CAPKP**2
        CON2=(SLP2-SLP1)/(2.*CAPKP**2)
        CON1=CON1-CON2
        CON3=(SLP2-SLP1)/(2.*CAPKP**4)
        CON4=1.5*(1.-SLP1)/(CAPKP**4)
        CON3=CON3-CON4
C       END


C        CONCENTRATE THE SURFACE CONTOURS (COMPUTATIONAL COORD ET)

        DO 244 I3=1,ZN2M1
        CAPFAC=0.
        ZETER1=ZET(I3)
244     ZET(I3)=SLP1*ZETER1+CON1*ZETER1**3+CON3*ZETER1**5-CAPFAC
241     CON5=2.5*(1.-SLP3)/CAPK**2
        CON6=(SLP4-SLP3)/(2.*CAPK**2)
        CON5=CON5-CON6
        CON7=(SLP4-SLP3)/2/CAPK**4
        CON8=1.5*(1.-SLP3)/CAPK**4
        CON7=CON7-CON8
        DO 248 J3=1,ENOPLS
        IF(J3.GT.ENO)ET(J3)=2.*CAPK-ET(J3)
        ET(J3)=SLP3*ET(J3)+CON5*ET(J3)**3+CON7*ET(J3)**5
        IF(J3.GT.ENO)ET(J3)=2.*CAPK-ET(J3)
248     CONTINUE
        SLP1=-SLP1
        IF(DELTA.NE.0.)SLP2=-DELTA
4446    FORMAT(6E13.5)
C       END




C       END OF CONCENTRATION PROCEDURE



205     X(1,1)=XUPS
        X(ZNO,1)=XDNS
        X(ZN2M1,1)=XUPS
        NZNOM1=ZNO-1


C       DO (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y UPPER SURFACE)
```

```
C     PROCEDURE (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y UPPER SURFACE)
      DO 260 I=1,ZNO
      DO 260 J=1,ENOPLS
      IF((I.EQ.1.OR.I.EQ.ZNO).AND.J.EQ.1) GOTO260
      ETER=ET(J)


C     DO (YOUR BASIC TRANSFORMATION )
C     PROCEDURE (YOUR BASIC TRANSFORMATION )
      ZED=ZET(I)
      IF(ETER.GT.CAPK)ZED=-ZED
      IF(ETER.GT.CAPK) ETER=2.*CAPK-ETER
      IF(I.EQ.ZNO)ZED=0.
      ALLFNC=FUDS(ETER, ZED,M1)
      LAM  = BADZAC(M1)
      TH   = OWOW(M1)
      X(I,J)= CHORD*A/PI*(LNK-2.*ALOG(ALLFNC/LAM))
      IF(J.GT.ENO.AND.I.EQ.ZN2M1)X(I,J)=-CHORD-X(I,J)
      IF(J.GT.ENO.AND.I.EQ.ZNO)X(I,J)=CHORD-X(I,J)
      Y(I,J)= 2.*A*CHORD/PI*TH
      IF(X(I,J).GT.XDNS)X(I,J)=XDNS
      IF(X(I,J).LT.XUPS)X(I,J)=XUPS
C     END

260   CONTINUE
C     END

      ZNP1=ZNO+1
      FK=1./SQRT(K)
      NZN2M2=ZN2M1-1


C     DO (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y LOWER SURFACE)

C     PROCEDURE (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y LOWER SURFACE)
      DO 271 I=ZNP1,ZN2M1
      IDIF=I-ZNO
      X(I,1)=X(ZNO-IDIF,1)-S*SIN(LAMDAO)
      Y(I,1)= -A*CHORD
      DO 271 J=1,ENOPLS
      IF(I.EQ.ZN2M1.AND.J.EQ.1) GOTO271
      ETER=ET(J)

C     DO (YOUR BASIC TRANSFORMATION )
C     PROCEDURE (YOUR BASIC TRANSFORMATION )
      ZED=ZET(I)
      IF(ETER.GT.CAPK)ZED=-ZED
      IF(ETER.GT.CAPK) ETER=2.*CAPK-ETER
```

```
      IF(I.EQ.ZNO)ZED=0.                                                QS028330
      ALLFNC=FUDS(ETER, ZED,M1)                                         QS028340
      LAM  = BADZAC(M1)                                                 QS028350
      TH   = OWOW(M1)                                                   QS028360
      X(I,J)= CHORD*A/PI*(LNK-2.*ALOG(ALLFNC/LAM))                      QS028370
      IF(J.GT.ENO.AND.I.EQ.ZN2M1)X(I,J)=-CHORD-X(I,J)                   QS028380
      IF(J.GT.ENO.AND.I.EQ.ZNO)X(I,J)=CHORD-X(I,J)                      QS028390
      Y(I,J)= 2.*A*CHORD/PI*TH                                          QS028400
      IF(X(I,J).GT.XDNS)X(I,J)=XDNS                                     QS028410
      IF(X(I,J).LT.XUPS)X(I,J)=XUPS                                     QS028420
C     END                                                               QS028430
                                                                        QS028440
271   CONTINUE                                                          QS028450
C     END                                                               QS028460
                                                                        QS028470
                                                                        QS028480
C     DO (SPECIAL POINTS)                                               QS028490
                                                                        QS028500
C     PROCEDURE (SPECIAL POINTS)                                        QS028510
      DO 278 J=1,ENOPLS                                                 QS028520
      X(1,J)=X(ZN2M1,J)                                                 QS028530
      Y(ZN2M1,J)=0.                                                     QS028540
      Y(ZNO,J)   =0.                                                    QS028550
278   Y(1,J)   =0.                                                      QS028560
C     END                                                               QS028570
                                                                        QS028580
                                                                        QS028590
      IF(STAG.EQ.0.)  GOTO 209                                          QS028600
                                                                        QS028610
                                                                        QS028620
C     DO (CIRCLE TRANSFM. FROM ZERO STAGGER TO INPUT VALUES)            QS028630
                                                                        QS028640
C     PROCEDURE (CIRCLE TRANSFM. FROM ZERO STAGGER TO INPUT VALUES)     QS028650
      CALL  TOSTAG(S,ZET,CHORD)                                         QS028660
      CHORD=CHORB-RTE/2.-RLE/2.                                         QS028670
      LAMDAO=STAG                                                       QS028680
      SOLIDO=SOLFLT                                                     QS028690
      S=CHORD/SOLFLT                                                    QS028700
      OLDK=CAPK                                                         QS028710
      OLDKP=CAPKP                                                       QS028720
                                                                        QS028730
C     DO (CALCULATION OF GEOMETRY CONSTANTS)                            QS028740
C     PROCEDURE (CALCULATION OF GEOMETRY CONSTANTS)                     QS028750
      A= COS(LAMDAO)/2./SOLIDO                                          QS028760
      K= EXP(-PI/2./A)                                                  QS028770
      LNK = -ALOG(K)                                                    QS028780
      M=K*K                                                             QS028790
      DM=M                                                              QS028800
```

```
      DM1=1.D0-DM
      M1=DM1
      CAPK=MMDELK(1,DM,IER)
      CAPKP=MMDELK(1,DM1,IER)
C     END
      DO 283 I1=1,ENOPLS
283   ET(I1)=ET(I1)*CAPK/OLDK
      DO 287 I1=1,ZN2M1
287   ZET(I1)=ZET(I1)*CAPKP/OLDKP
C     END

209   CONTINUE

C     DO (TRANSLATE ORIGIN OF INPUT BODY POINTS TO MIDCHORD)

C     PROCEDURE (TRANSLATE ORIGIN OF INPUT BODY POINTS TO MIDCHORD)
      DO 291 L=1,END
291   XBOD(L)=XBOD(L)-RLE/2.-.5*CHORD
C     END

C     DO (TRAILING EDGE FIX OF CAMBER POINTS)

C     PROCEDURE (TRAILING EDGE FIX OF CAMBER POINTS)
      ENOP1=ENO+1
      DO 295 J=ENOP1,ENOPLS
      JNEXT=2*ENO-J
      FAD=1.
      IF(X(ZNO+1,ENO)-X(ZNO-1,ENO).NE.0.)
     1FAD=(X(ZNO,ENO)-X(ZNO-1,ENO))/(X(ZNO+1,ENO)-X(ZNO-1,ENO))
      FAB=1.
      IF(X(2,ENO)-X(ZN2M1-1,ENO).NE.0.)
     1FAB=(X(1,ENO)-X(ZN2M1-1,ENO))/(X(2,ENO)-X(ZN2M1-1,ENO))
      X(ZNO,J)=X(ZNO-1,J)+FAD*(X(ZNO+1,J)-X(ZNO-1,J))
      Y(ZNO,J)=Y(ZNO-1,J)+FAD*(Y(ZNO+1,J)-Y(ZNO-1,J))
      X(1,J)=X(ZN2M1-1,J)+FAB*(X(2,J)-X(ZN2M1-1,J))
      Y(1,J)=Y(ZN2M1-1,J)+FAB*(Y(2,J)-Y(ZN2M1-1,J))
      X(ZN2M1,J)=X(1,J)
295   Y(ZN2M1,J)=Y(1,J)
C     END


C     DO (ADDITION OF BLADE SURFACE ON CLEAN MESH)
      IF(DOSURF)  CALL SURFUP(ET,ZET,XBOD,YBOD,EF)
```

```
7777  FORMAT(14E9.3)
      RETURN
      END
      DOUBLE PRECISION FUNCTION MMDELK(ILPT,ARG,IER)
C-----CALCS COMPLETE ELLIPTIC INTEGRAL OF FIRST KIND FOR MODULUS SQUARED
C      = ARG
      DOUBLE PRECISION ARG,PI,A,B,C,AO,BO
      DATA PI/3.14159265358979300/
      IER = 130
      IF(ILPT.NE.1) STOP 105
      IF(ARG.LT.0.DO.OR.ARG.GE.1.DO) GOTO40
      IER = 0
      A = 1.CO
      B = DSQRT(1.DO - ARG)
      DO 10 I = 1,15
      C = .5DO * (A - B)
      IF(DABS(C).LT.1.D-12) GOTO15
      AO = A
      BO = B
      A = .5DO * (AO + BO)
      B = DSQRT(AO * BO)
10    CONTINUE
15    IF(I.GT.15) STOP 110
      MMDELK = .5DO * PI / A
40    RETURN
      END
      SUBROUTINE SURFUP(ET,ZET,XBOD,YBOD,EF)
```

```
C           LOCATES INTERSECTION OF EACH RADIATING LINE IN THE FLATPLATE
C       GRID WITH
C           THE DESIRED BODY SHAPE. ADJUSTS THE SURFACE CONTOURS ('ETA VA
C       LUE STRETCH')
C           SO THAT ALL RADIATING GRID LINES END ON THE BLADE.

C       ET(ENO)=CAPK
C       DELSMX=.06667
C       PIO2=1.57079
C       THETT=THETT*PI/180.
C       THETL=THETL*PI/180.
C       ISA=1
C       PLATE=(CHORD-RLE/2.-RTE/2.)/2.
C       DO (BODY COORDINATE LOADING)
C       ZN2M1=2*ZNO-1
C       NZ2M2=ZN2M1-1
C       DO FOR I=1,NZ2M2
C          ENPOLS=ENOPLS
C          DO (ITERATION SETUP)
C          DO (ITERATION FOR ETA LINE-BODY INTERSECTION)
C          DO FOR J=1,ENOPLS
C             IF(ET(J).GE.ETA) THEN
C             EXIT(J)
C             END
C          END
C          IF(J.EQ.ENOPLS+1) ETA=ET(ENOPLS)
C          IF(RELAXD) THEN
C             XX(I,ENO)=X
C             YY(I,ENO)=Y
C             ETS(I)=ETA
C             DO (ETA VALUE STRETCH)
C          END
C       END
C       DO (EDGE REGIONS)
C       DO FOR J1=2,ENO
C          XX(ZN2M1,J1)=XX(1,J1)
C          YY(ZN2M1,J1)=YY(1,J1)
C       END
C       RETURN
```

```
      DOUBLE PRECISION CAPK,CAPKP
      COMMON/GEOM/ZNO,ENO,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK
     1,CAPKP,PI
     *,LNK,M1,KM,END,M,DOSURF,AARR,SOL,ENOPLS
     1,SDUM(5),SMOOTH,CHOP,THETT,THETL,TDUM,LEONLY
      COMMON /ENTIRE/XX(100,30),YY(100,30)
      COMMON/SPGENC/DA,DSMAX,RMAX,THIN,DB,TMIN,DSEND
      COMMON/FUNCOS/S,C,D,SN,CN,DN
      COMMON/MAN/DELSMX,PIO2,DELS1,IHUB
      DIMENSION XBOD(200),YBOD(200),SU(400),SD(400),ET(100),ZET(100),
     1NSAVE(25),SEDGT(400),SEDGL(400),
     1ISAVE(25),ETS(100),SYYETA(400),SDXETA(400),SXYETA(400),SDYETA(400)
      INTEGER ZNO,ENO,END,ZN2M1,ENOPLS,ENDM1,ENPOLS,ENDMM1,ENM1
      LOGICAL DUM,RZTRP,IZTRP,THIN,DONE,SMOOTH,DOSURF,LEONLY
      LOGICAL TE,RELAXD,BOTTOM/.FALSE./,FAILD/.FALSE./,EDGES/.FALSE./
      COMPLEX FZTRP,DZTRP,FZ,  Z(400),Z2(400),XOETA(400),YOETA(400)
     1,EDGL(400),EDGT(400),XYETA(400),YYETA(400)
      REAL M,M1,LNK,LAMDAO
      NAMELIST/SPGENS/DA,DSMAX,RMAX,THIN,DB,TMIN,DSEND
      NAMELIST/TESY/I,IS,ETA,Y,X,SPX,SPY,SPE,ITST,DF1DN,DF2DN,DF3DX,
     1F1,F2,F3,DELN,DELX,DELY,FZ,SP
      NAMELIST/TESX/ XYETA,SXYETA,YYETA,SYYETA
      DATA Z/400*(1.E20,0.)/
      DATA Z2/400*(1.E20,0.)/
      DATA EDGT,EDGL,XOETA,YOETA,XYETA,YYETA/2400*(1.E20,0.)/


      ET(ENO)=CAPK
      DELSMX=.06667
      PIO2=1.57079
      THETT=THETT*PI/180.
      THETL=THETL*PI/180.
      ISA=1
      PLATE=(CHORD-RLE/2.-RTE/2.)/2.


C     DO (BODY COORDINATE LOADING)

C     PROCEDURE (BODY COORDINATE LOADING)

      IF(.NOT.SMOOTH) GOTO222
```

```
      IF(LEONLY) GOTO1008                                               QS030730
                                                                        QS030740
      KMH=(KM+1)/2                                                      QS030750
      IF(MOD(KM,2).NE.0)KMH=KMH-1                                       QS030760
      DO 234 I1=1,KMH                                                   QS030770
      I5=I1+KMH                                                         QS030780
234   Z(I1)=CMPLX(XBOD(I5),YBOD(I5))                                    QS030790
      IF(MOD(KM,2).NE.0)                                                QS030800
     1Z(KMH+1)=CMPLX(XBOD(I5+1),YBOD(I5+1))                             QS030810
      IF(MOD(KM,2).NE.0)KMH=KMH+1                                       QS030820
      KWOW=KMH+1+END-KM-2-(KM+1)/2+KM/2                                 QS030830
      KMH1=KMH+1                                                        QS030840
      DO 240 I1=KMH1,KWOW                                               QS030850
      I5=END-I1+KMH                                                     QS030860
240   Z(I1)=CMPLX(XBOD(I5),YBOD(I5))                                    QS030870
      KWOWP1=KWOW+1                                                     QS030880
      ENDMM1=END-1                                                      QS030890
      DO 244 I1=KWOWP1,ENDMM1                                           QS030900
      I5=I1-END+(KM-1)/2+3-MOD(KM,2)                                    QS030910
244   Z(I1)=CMPLX(XBOD(I5),YBOD(I5))                                    QS030920
      GOTO225                                                           QS030930
1008  KMH=END-KM                                                        QS030940
      DO 226 I1=1,KMH                                                   QS030950
      I5=END-I1+1                                                       QS030960
226   Z(I1)=CMPLX(XBOD(I5),YBOD(I5))                                    QS030970
      KWOW=END-1                                                        QS030980
      KMH1=KMH+1                                                        QS030990
      DO 230 I1=KMH1,KWOW                                               QS031000
      I5=I1-KMH+1                                                       QS031010
230   Z(I1)=CMPLX(XBOD(I5),YBOD(I5))                                    QS031020
4447  FORMAT(6E13.5)                                                    QS031030
225   ENDM1=END-1                                                       QS031040
C     WRITE(6,1234) (Z(IW),IW=1,ENDM1),                                 QS031050
C    1(XBOD(IQ),YBOD(IQ),IQ=1,END)                                      QS031060
1234  FORMAT(10E13.6)                                                   QS031070
                                                                        QS031080
                                                                        QS031090
      DA=.17                                                            QS031100
C     WRITE(6,1235)                                                     QS031110
1235  FORMAT(55H0THE FOLLOWING IS A PRINT   OF THE Z MATRIX FROM SURFUP)QS031120
C     WRITE(6,*)Z                                                       QS031130
      CALL XYCALC(1,ENDM1,Z,SU,SD)                                      QS031140
C     WRITE(6,1236)                                                     QS031150
1236  FORMAT(69H0THE FOLLOWING IS A PRINT   OF Z,SU,SD MATRICES AFTER THQS031160
     +E CALL XYCALC)                                                    QS031170
C     WRITE(6,*)Z                                                       QS031180
C     WRITE(6,*)SU                                                      QS031190
C     WRITE(6,*)SD                                                      QS031200
```

```
      END=ENDM1                                                         QS031210
      ITS=0.                                                            QS031220
      XTS=0.                                                            QS031230
      DO 248 I1=1,END                                                   QS031240
      IF(SU(I1).GT.XTS) ITS=I1                                          QS031250
      IF(SU(I1).GT.XTS) XTS=SU(I1)                                      QS031260
  248 CONTINUE                                                          QS031270
      KMP=ITS                                                           QS031280
      ITS=0                                                             QS031290
      XTS=100.                                                          QS031300
      DO 254 I1=1,END                                                   QS031310
      IF(SU(I1).LT.XTS) ITS=I1                                          QS031320
      IF(SU(I1).LT.XTS) XTS=SU(I1)                                      QS031330
  254 CONTINUE                                                          QS031340
      K1P=ITS                                                           QS031350
      KM=END-(K1P-KMP)                                                  QS031360
      KM1=KM+1                                                          QS031370
      END=END+1                                                         QS031380
      DO 260 I1=1,KM                                                    QS031390
      I5=K1P+I1-1                                                       QS031400
      IF(I5.GE.END-1)I5=I1-END+K1P+1                                    QS031410
      XBOD(I1)=SU(I5)                                                   QS031420
  260 YBOD(I1)=SD(I5)                                                   QS031430
 4448 FORMAT(4E13.5,3I7)                                                QS031440
                                                                        QS031450
C      LOAD SMOOTHED DATA AROUND L.E. REGION INTO EDGL FOR USE IN 'EDGE QS031460
C     REGION' TECHNIQUE                                                 QS031470
                                                                        QS031480
      DO 264 I1=KM1,END                                                 QS031490
      I5=END+KMP-I1                                                     QS031500
      XBOD(I1)=SU(I5)                                                   QS031510
  264 YBOD(I1)=SD(I5)                                                   QS031520
      DO 268 I1=KMP,K1P                                                 QS031530
      IF(SU(I1).LT.-CHOP*.50*PLATE) GOTO271                             QS031540
  268 CONTINUE                                                          QS031550
      I1=K1P                                                            QS031560
  271 I2L=0                                                             QS031570
      DO 274 I3=I1,END                                                  QS031580
      IF(ABS(SU(I3)).LT.CHOP*.50*PLATE) GOTO277                         QS031590
      YEE=SU(I3)*SIN(THETL)+SD(I3)*COS(THETL)                           QS031600
      I2L=I2L+1                                                         QS031610
  274 EDGL(I2L)=CMPLX(YEE,SU(I3))                                       QS031620
  277 NOTO=SGEN(SEDGL,EDGL,I2L)                                         QS031630
                                                                        QS031640
C      IF (.NOT.LEONLY) LOAD SMOOTHED DATA AROUND T.E. INTO EDGT FOR USEQS031650
C      IN 'EDGE REGION'                                                 QS031660
                                                                        QS031670
      IF(LEONLY) GOTO280                                                QS031680
```

```
      DO 282 I1=1,KMP
      IF(SU(I1).GT.CHOP*.50*PLATE) GOTO285
282   CONTINUE
285   I2T=0
      DO 288 I3=I1,K1P
      IF(SU(I3).LT.CHOP*.50*PLATE) GOTO291
      YEE=SU(I3)*SIN(THETT)+SD(I3)*COS(THETT)
      I2T=I2T+1
288   EDGT(I2T)=CMPLX(YEE,SU(I3))
291   NOTO=SGEN(SEDGT,EDGT,I2T)


280   DO 298 I1=1,400
      Z(I1)=1.E20
      IF(I1.LE.400) SU(I1)=0.
      IF(I1.LE.400)SD(I1)=0.
298   CONTINUE
222   DO 2100 I1=1,KM
2100  Z(I1)=CMPLX(XBOD(I1),YBOD(I1))
      NOTO= SGEN(SU,Z,KM)
      KM1=KM+1
      IF(NOTO.NE.0) WRITE(6,2) NOTO
2     FORMAT('OSGEN FAILURE',I6)
      KENDO=END-KM
      DO 2104 I1=KM1,END
      I2=I1-KM
2104  Z2(I2)=CMPLX(XBOD(I1),YBOD(I1))
      NOTO=SGEN(SD,Z2,KENDO)
C     END OF BODY COORDINATE LOADING

      ZN2M1=2*ZNO-1
      NZ2M2=ZN2M1-1


      DO 202 I=1,NZ2M2
      ENPOLS=ENOPLS

C     DO (ITERATION SETUP)

C     PROCEDURE (ITERATION SETUP)
      ENPOLS=ENOPLS
      RELAXD=.FALSE.
      FAILD=.FALSE.
      EDGES=.FALSE.
      BOTTOM=.FALSE.
```

```
C       DO (GRID-LINE COORDINATE LOADING)

C       PROCEDURE (GRID-LINE COORDINATE LOADING)
        ITST=0
        JTST=0
        DO 20109 I1=1,400
        XOETA(I1)=1.E20
        YOFTA(I1)=1.E20
        IF(I1.LE.400) SDXETA(I1)=0.
        IF(I1.LE.400)SDYETA(I1)=0.
20109   CONTINUE
        DO 20115 I1=1,ENPOLS
        IF(ABS(XX(I,1)).GT.20.)XX(I,1)=SIGN(7.*ABS(XX(I,2)),XX(I,1))
        XOETA(I1)=CMPLX(ET(I1),XX(I,I1))
20115   YOETA(I1)=CMPLX(ET(I1),YY(I,I1))
        NOTO=SGEN(SDYETA,YOETA,ENPOLS)
        IF(NOTO.NE.0) WRITE(6,2) NOTO
        NOTO=SGEN(SDXETA,XOETA,ENPOLS)
        IF(NOTO.NE.0) WRITE(6,2) NOTO
C       END

        Y=YY(I,ENO)
        IF(Y.EQ.0.)Y=SIGN(1.E-04,YY(I,ENO-1))
        X=XX(I,ENO-1)
        XOO=X
        ETA= CAPK*(ENO-2)/(ENO-1)
        IF(Y.LE.0..AND.I.NE.1)BOTTOM=.TRUE.
        IF(I.EQ.ZNO.AND.LAMDAO.GT.0.)BOTTOM=.TRUE.
        DF1DN=(XX(I,ENO)-XX(I,ENO-1))/(ET(ENO-1)-ET(ENO))
        DF2DN=(YY(I,ENO)-YY(I,ENO-1))/(ET(ENO-1)-ET(ENO))
        DF3DX=0.
        IF(DF2DN+DF1DN*DF3DX.EQ.0.)DF3DX=1.
        DELN=0.
        DELX=0.
        DELY=0.
        SP=0.
C       END

C       DO (ITERATION FOR ETA LINE-BODY INTERSECTION)

C       PROCEDURE (ITERATION FOR ETA LINE-BODY INTERSECTION)
        IF(ABS(XX(I,ENO)).GT.CHOP*PLATE.AND.
       1((.NOT.LEONLY.OR.I.LE.ZNO/2.OR.I.GE.3*ZNO/2))) GOTO2119
C       DO UNTIL(RELAXD.OR.FAILD)
2122    ITST=ITST+1
        X=X+DELX
```

```
      Y=Y+DELY
      SPY=0.
      SPX=0.
      SP=0.
      SPE=0.
      ETA=ETA+DELN
      IF(ETA.LT.0.)ETA=ET(2)
      IF(ETA.GT.2.*CAPK)ETA=ET(ENOPLS)
      IF(.NOT.EDGES) GOTO2124
      IF(ETA.GT..90*CAPK.AND.ITST.GT.20)ETA=.90*CAPK
      DUM=RZTRP(SXYETA,XYETA,ETA,I2S,SPX)
      GETA=AIMAG(FZTRP(SXYETA,XYETA,SPX,I2S))
      DUM=RZTRP(SYYETA,YYETA,ETA,I2Y,SPY)
      FETA=AIMAG(FZTRP(SYYETA,YYETA,SPY,I2Y))
      GOTO2125
2124  DUM=RZTRP(SDXETA,XOETA,ETA,ENPOLS,SPX)
      GETA=AIMAG(FZTRP(SDXETA,XOETA,SPX,ENPOLS))
      DUM=RZTRP(SDYETA,YOETA,ETA,ENPOLS,SPY)
      FETA=AIMAG(FZTRP(SDYETA,YOETA,SPY,ENPOLS))
2125  F1=X-GETA
      F2=Y-FETA
      IF(.NOT.BOTTOM) GOTO2126
      DUM=RZTRP(SD,Z2,X,KENDO,SP)
      FX=AIMAG(FZTRP(SD,Z2,SP,KENDO))
      GOTO2127
2126  DUM=RZTRP(SU,Z,X,KM,SP)
      FX=AIMAG(FZTRP(SU,Z,SP,KM))
2127  F3 = Y-FX
      RELAXD=(ABS(F1)+ABS(F2)+ABS(F3)).LT..0001
C      IF FIRST TRY IS UNSUCCESSFUL, GET SMOOTHED SLOPE DATA AND TRY AGA
C     IN
      IF((ITST.LE.20.AND.ABS(DELN/ETA).LE.1000.).OR.EDGES) GOTO2128
C     DO (ITERATION WITH BETTER SLOPES)
C     PROCEDURE(ITERATION WITH BETTER SLOPES)
      WRITE(6,5) I
5     FORMAT('OITERATION PROCEEDS WITH SMOOTHER SLOPE DATA FOR LINE ',I4
     1)
      DO 2139 I1=1,400
      XYETA(I1)=1.E20
      YYETA(I1)=1.E20
      IF(I1.LE.400) SXYETA(I1)=0.
      IF(I1.LE.400)SYYETA(I1)=0.
2139  CONTINUE
C     DO (ITERATION SETUP AGAIN)
```

```
C       PROCEDURE (ITERATION SETUP)
        ENPOLS=ENOPLS
        RELAXD=.FALSE.
        FAILD=.FALSE.
        EDGES=.FALSE.
        BOTTOM=.FALSE.

C       DO (GRID-LINE COORDINATE LOADING AGAIN)
C       PROCEDURE (GRID-LINE COORDINATE LOADING)
        ITST=0
        JTST=0
        DO 2109 I1=1,400
        XOETA(I1)=1.E20
        YOETA(I1)=1.E20
        IF(I1.LE.400) SDXETA(I1)=0.
        IF(I1 .LE.400)SDYETA(I1)=0.
2109    CONTINUE
        DO 21115 I1=1,ENPOLS
        IF(ABS(XX(I,1)).GT.20.)XX(I,1)=SIGN(7.*ABS(XX(I,2)),XX(I,1))
        XOETA(I1)=CMPLX(ET(I1),XX(I,I1))
21115   YOETA(I1)=CMPLX(ET(I1),YY(I,I1))
        NOTO=SGEN(SDYETA,YOETA,ENPOLS)
        IF(NOTO.NE.0) WRITE(6,2) NOTO
        NOTO=SGEN(SDXETA,XOETA,ENPOLS)
        IF(NOTO.NE.0) WRITE(6,2) NOTO
C       END

        Y=YY(I,ENO)
        IF(Y.EQ.0.)Y=SIGN(1.E-04,YY(I,ENO-1))
        X=XX(I,ENO-1)
        XOO=X
        ETA= CAPK*(ENO-2)/(ENO-1)
        IF(Y.LE.0..AND.I.NE.1)BOTTOM=.TRUE.
        IF(I.EQ.ZNO.AND.LAMDAO.GT.0.)BOTTOM=.TRUE.
        DF1DN=(XX(I,ENO)-XX(I,ENO-1))/(ET(ENO-1)-ET(ENO))
        DF2DN=(YY(I,ENO)-YY(I,ENO-1))/(ET(ENO-1)-ET(ENO))
        DF3DX=0.
        IF(DF2DN+DF1DN*DF3DX.EQ.0.)DF3DX=1.
        DELN=0.
        DELX=0.
        DELY=0.
        SP=0.
C       END

        EDGES=.TRUE.
        DO 2146 I6=1,ENO
C       DO 2146 I6=1,ENOPLS
        XYETA(I6)=XOETA(I6)
```

```
2146   YYETA(I6)=YOETA(I6)                                         QS033610
       DA=.02                                                      QS033620
       DELSMX=.06667                                               QS033630
C      I2S=ENOPLS                                                  QS033640
C       SMOOTH THE SLOPE DATA                                      QS033650
       CALL XYCALC(1,I2S,XYETA,SXYETA,SYYETA)                      QS033660
       DO 2150 I6=1,I2S                                            QS033670
2150   XYETA(I6)=CMPLX(SXYETA(I6),SYYETA(I6))                      QS033680
       DO 2154 I4=1,400                                            QS033690
       SXYETA(I4)=0.                                               QS033700
2154   SYYETA(I4)=0.                                               QS033710
C      I2Y=ENOPLS                                                  QS033720
       CALL XYCALC(1,I2Y,YYETA,SXYETA,SYYETA)                      QS033730
       DELSMX=.066667                                              QS033740
       DO 2158 I6=1,I2Y                                            QS033750
2158   YYETA(I6)=CMPLX(SXYETA(I6),SYYETA(I6))                      QS033760
       NOTO=SGEN(SXYETA,XYETA,I2S)                                 QS033770
       NOTO=SGEN(SYYETA,YYETA,I2Y)                                 QS033780
C      WRITE(6,TESX)                                               QS033790
C       END OF BETTER SLOPE ACQUISITION, EDGES NOW = .TRUE.        QS033800
2128   IF(ITST.GT.50.OR.ABS(DELN/ETA).GT.1000.) FAILD=.TRUE.       QS033810
       IF(RELAXD) GOTO2129                                         QS033820
       DELN = (F3-F2-DF3DX*F1)/(DF2DN+DF1DN*DF3DX)                 QS033830
       DELX=-F1-DF1DN*DELN                                         QS033840
       DELY = -F3 -DF3DX*DELX                                      QS033850
C      DO (SEARCH FOR BODY SLOPE)                                  QS033860
C      PROCEDURE (SEARCH FOR BODY SLOPE)                           QS033870
       K4=1                                                        QS033880
       K3=KM                                                       QS033890
       K2=1                                                        QS033900
       IF(.NOT.BOTTOM) GOTO2162                                    QS033910
       K2=KM+1                                                     QS033920
       K3=END                                                      QS033930
       K4=1                                                        QS033940
2162   DO 2164 K1=K2,K3,K4                                         QS033950
       IF(XBOD(K1).GE. X+DELX) GOTO2167                            QS033960
2164   CONTINUE                                                    QS033970
       K1=K3                                                       QS033980
2167   DF3DX=(YBOD(K1-1)-YBOD(K1))/                                QS033990
     *  (XBOD(K1)-XBOD(K1-1))                                      QS034000
C      END                                                         QS034010
                                                                   QS034020
C      DO (SEARCH FOR GRID-LINE SLOPE)                             QS034030
C      PROCEDURE (SEARCH FOR GRID-LINE SLOPE)                      QS034040
       IF(.NOT.EDGES) GOTO2170                                     QS034050
       DO 2172 J=1,I2S                                             QS034060
       IF(REAL(XYETA(J)).GE.ETA) GOTO2175                          QS034070
2172   CONTINUE                                                    QS034080
```

```
2175  IF(J.EQ.I2S+1) J=I2S                                        QS034090
      DO 2180 JY=1,I2Y                                            QS034100
      IF(REAL(YYETA(JY)).GE.ETA) GOTO2183                         QS034110
2180  CONTINUE                                                    QS034120
2183  IF(JY.EQ.I2Y+1) JY=I2Y                                      QS034130
      DF1DN=(AIMAG(XYETA(J))-AIMAG(XYETA(J-1)))/                  QS034140
     1(REAL(XYETA(J-1))-REAL(XYETA(J)))                           QS034150
      DF2DN=(AIMAG(YYETA(JY))-AIMAG(YYETA(JY-1)))/                QS034160
     1(REAL(YYETA(JY-1))-REAL(YYETA(JY)))                         QS034170
      GOTO2171                                                    QS034180
2170  DO 2188 J=1,ENOPLS                                          QS034190
      IF(ET(J).GE.ETA) GOTO2191                                   QS034200
2188  CONTINUE                                                    QS034210
2191  IF(J.EQ.ENOPLS+1) J=ENOPLS                                  QS034220
      DF1DN=(XX(I,J)-XX(I,J-1))/(ET(J-1)-ET(J))                   QS034230
      DF2DN=(YY(I,J)-YY(I,J-1))/(ET(J-1)-ET(J))                   QS034240
2171  CONTINUE                                                    QS034250
C     END                                                         QS034260
                                                                  QS034270
2129  IF(.NOT.(DF2DN+DF3DX.EQ.0..OR.                              QS034280
     1((I.EQ.1.OR.I.EQ.ZNO).AND.LAMDAO.EQ.0.).OR.                 QS034290
     1(.NOT.SMOOTH.AND.ABS(XX(I,ENO)).GT..98*PLATE.AND.FAILD))) GOTO2121QS034300
      RELAXD=.TRUE.                                               QS034310
      FAILD=.FALSE.                                               QS034320
      Y=YY(I,ENO)                                                 QS034330
      X=XBOD(KM)                                                  QS034340
      IF(I.EQ.1)X=XBOD(1)                                         QS034350
      IF(SMOOTH) GOTO2135                                         QS034360
      Y=0.                                                        QS034370
      X=XBOD(1)                                                   QS034380
      IF(I.GT.ZNO/2.AND.I.LT.3*ZNO/2)X=XBOD(KM)                   QS034390
2135  DUM=IZTRP(SDXETA,XOETA,X,ENO,SPE)                           QS034400
      ETA=REAL(FZTRP(SDXETA,XOETA,SPE,ENO))                       QS034410
                                                                  QS034420
2121  IF(.NOT.(RELAXD.OR.FAILD))GOTO2122                          QS034430
C     END OF ITERATION FOR ETA LINE-BODY INTERSECTION             QS034440
2119  IF(RELAXD.OR.ABS(XX(I,ENO)).LE.CHOP*PLATE.AND..NOT.FAILD) GOTO3004QS034450
      IF(FAILD)WRITE(6,3) I                                       QS034460
      IF(.NOT.FAILD)WRITE(6,4) I,XX(I,ENO),PLATE                  QS034470
3     FORMAT('0ITERATION FAILD FOR LINE ',I4,                     QS034480
     1' PROCEEDING TO EDGE REGION TECHNIQUE')                     QS034490
4     FORMAT('0ITERATION NOT ATTEMPTED FOR LINE ',I4,             QS034500
     1', BODY X TOO CLOSE TO LE OR TE: ',2E15.6,'PROCEED TO EDGE TECH.')QS034510
      ISAVE(ISA)=I                                                QS034520
      NSAVE(ISA)=1                                                QS034530
      IF(BOTTOM)NSAVE(ISA)=-1                                     QS034540
      ISA=ISA+1                                                   QS034550
3004  DO 208 J=1,ENOPLS                                           QS034560
```

```
      IF(ET(J).GE.ETA) GOTO211
208   CONTINUE
211   IF(J.EQ.ENOPLS+1) ETA=ET(ENOPLS)
      IF(.NOT.RELAXD) GOTO202
      XX(I,ENO)=X
      YY(I,ENO)=Y
      ETS(I)=ETA
C     DO (ETA VALUE STRETCH)
C     PROCEDURE (ETA VALUE STRETCH)
      ENM1=ENO-1
      DO 20196 J=2,ENM1
      ETANEW=ET(J)/CAPK*ETA
      SPX=SDXETA(1)
      SPY=SDYETA(1)
      DUM=RZTRP(SDXETA,XOETA,ETANEW,ENPOLS,SPX)
      XX(I,J)=AIMAG(FZTRP(SDXETA,XOETA,SPX,ENPOLS))
      DUM=RZTRP(SDYETA,YOETA,ETANEW,ENPOLS,SPY)
20196 YY(I,J)=AIMAG(FZTRP(SDYETA,YOETA,SPY,ENPOLS))
C     END

202   CONTINUE
C     DO (EDGE REGIONS)
C     PROCEDURE (EDGE REGIONS)
C
C 'EDGE REGION TECHNIQUE' = INTERPOLATE ON CURVE OF INTERSECTION POINTS
C                           DEFINED BY SUCCESSFUL NEIGHBORING LINES.
C
      NOZO2=ZNO/2
      IF(ISAVE(1).GE.NOZO2) GOTO2020
      DO 2202 I3=1,ISA
      IF(ISAVE(I3).GT.NOZO2) GOTO2205
2202  CONTINUE
      I3=ISA
2205  I3=I3-1
      ISAP=ISA
      DO 2208 I4=1,I3
      ISAVE(ISAP)=ISAVE(I4)
2208  ISAP=ISAP+1
      ISAM1=ISA-1
      DO 2212 I4=1,ISAM1
      I4I3=I4+I3
2212  ISAVE(I4)=ISAVE(I4I3)
2020  DO 2216 I1=1,400
      XYETA(I1)=1.E20
      YYETA(I1)=1.E20
      IF(I1.GT.400) GOTO2216
      SXYETA(I1)=0.
```

```
      SYYETA(I1)=0.
      SU(I1)=0.
      SD(I1)=0.
 2216 CONTINUE
      I2=0
      IS=1
      NZ2M11=ZN2M1-1

C      LOAD VALUES OF BODY Y AND ZET FROM  ALL SUCCESSFUL LINE ITERATION
C     S INTO
C      THE ARRAY XYETA AND SMOOTH IT.
      DO 2222 I1=NOZO2,NZ2M11
      IF(IS.GT.ISA) GOTO2222
      IF(ISAVE(IS).EQ.I1) GOTO2228
      I2=I2+1
      THE=XX(I1,ENO)*(THETL-THETT)/(-1.6*CHOP*PLATE)
      IF(XX(I1,ENO).LT.-CHOP*.80*PLATE)THE=THETL
      IF(XX(I1,ENO).GT.CHOP*.80*PLATE)THE=THETT
      YEE=YY(I1,ENO)*COS(THE)+XX(I1,ENO)*SIN(THE)
      YEE=YEE*10.
      XYETA(I2)=CMPLX(ZET(I1),YEE)
      YYETA(I2)=CMPLX(ZET(I1),ETS(I1))
 2228 IF(ISAVE(IS).EQ.I1)IS=IS+1
 2222 CONTINUE
      DO 2230 I1=1,NOZO2
      IF(ISAVE(IS).EQ.I1) GOTO2234
      I2=I2+1
      ZED=ZET(I1)-2.*CAPKP
      THE=XX(I1,ENO)*(THETL-THETT)/(-1.6*PLATE)
      IF(XX(I1,ENO).LT.-CHOP*.80*PLATE)THE=THETL
      IF(XX(I1,ENO).GT.CHOP*.80*PLATE)THE=THETT
      YEE=YY(I1,ENO)*COS(THE)+XX(I1,ENO)*SIN(THE)
      YEE=YEE*10.
      XYETA(I2)=CMPLX(ZED,YEE)
      YYETA(I2)=CMPLX(ZED,ETS(I1))
 2234 IF(ISAVE(IS).EQ.I1)IS=IS+1
 2230 CONTINUE
      I2S=I2
      DELSMX=.5
      DA=.05
      CALL XYCALC(1,I2S,XYETA,SU,SD)
      DELSMX=.066667
      DO 2236 I6=1,I2S
 2236 XYETA(I6)=CMPLX(SU(I6),SD(I6))
      NOTO=SGEN(SXYETA,XYETA,I2S)
      NOTO=SGEN(SYYETA,YYETA,I2)
C     WRITE(6,TESX)
      SPT=0.
```

```
      SPL=0.
      ISAM1=ISA-1
      DO 2240 IS=1,ISAM1
      SPX=0.
      SPE=0.
      TE=.FALSE.
      I1=ISAVE(IS)
      I=I1
      IMM1=I-1
      IF(IMM1.EQ.0)IMM1=ZN2M1-1


      IF(I.GT.ZN2M1/4.AND.I.LT.3*ZN2M1/4)TE=.TRUE.


      IF(TE)SPL=0.
      ZED=ZET(I1)

C      INTERPOLATE ON THE TABLE OF Y VALUES IN XYETA USING THE KNOWN VAL
C     UE OF ZETA
C      FOR EACH UNSUCCESSFUL LINE.

      IF(.NOT.TE.AND.ZED.GT..85*CAPKP)ZED=ZED-2.*CAPKP
      DUM=RZTRP(SXYETA,XYETA,ZED,I2S,SPX)
      Y=AIMAG(FZTRP(SXYETA,XYETA,SPX,I2S))/10.


C      AFTER Y IS FOUND, OBTAIN X FROM TABLE OF L.E. AND T.E. BODY SHAPE
C     S
C      IN EDGL AND EDGT.


      IF(.NOT.SMOOTH) GOTO2244
      IF(.NOT.TE) GOTO2246
      DUM=RZTRP(SEDGT,EDGT,Y,I2T,SPT)
      X=AIMAG(FZTRP(SEDGT,EDGT,SPT,I2T))
      IF(ABS(X-XX(IMM1,ENO)).LE.1.5/FLOAT(ZNO)) GOTO22461
      DUM=RZTRP(SEDGT,EDGT,Y,I2T,SPT)
      X=AIMAG(FZTRP(SEDGT,EDGT,SPT,I2T))
22461 GOTO2245
2246  DUM=RZTRP(SEDGL,EDGL,Y,I2L,SPL)
      X=AIMAG(FZTRP(SEDGL,EDGL,SPL,I2L))
      IF(ABS(X-XX(IMM1,ENO)).LE.1.5/FLOAT(ZNO)) GOTO22441
      DUM=RZTRP(SEDGL,EDGL,Y,I2L,SPL)
      X=AIMAG(FZTRP(SEDGL,EDGL,SPL,I2L))
22441 GOTO2245
2244  IF(Y.GT.0..OR.I.EQ.1) GOTO2252
      DUM=IZTRP(SD,Z2,Y,KENDO,SP)
```

```
      X=REAL(FZTRP(SD,Z2,SP,KENDO))
      GOTO2245
 2252 DUM=IZTRP(SU,Z,Y,KM,SP)
      X=REAL(FZTRP(SU,Z,SP,KM))


 2245 IF(.NOT.TE)Y=(Y-X*SIN(THETL))/COS(THETL)
      IF(TE)Y=(Y-X*SIN(THETT))/COS(THETT)
C     DO (GRID-LINE COORDINATE LOADING)
C     PROCEDURE (GRID-LINE COORDINATE LOADING)
      ITST=0
      JTST=0
      DO 21109 I1=1,400
      XOETA(I1)=1.E20
      YOETA(I1)=1.E20
      IF(I1.LE.400) SDXETA(I1)=0.
      IF(I1.LE.400)SDYETA(I1)=0.
21109 CONTINUE
      DO 2115 I1=1,ENPOLS
      IF(ABS(XX(I,1)).GT.20.)XX(I,1)=SIGN(7.*ABS(XX(I,2)),XX(I,1))
      XOETA(I1)=CMPLX(ET(I1),XX(I,I1))
 2115 YOETA(I1)=CMPLX(ET(I1),YY(I,I1))
      NOTO=SGEN(SDYETA,YOETA,ENPOLS)
      IF(NOTO.NE.0) WRITE(6,2) NOTO
      NOTO=SGEN(SDXETA,XOETA,ENPOLS)
      IF(NOTO.NE.0) WRITE(6,2) NOTO
C     END
      DUM=IZTRP(SDXETA,XOETA,X,ENPOLS,SPE)
      ETA=REAL(FZTRP(SDXETA,XOETA,SPE,ENPOLS))


C     DO (ETA VALUE STRETCH FOR EDGE REGION POINTS)

C     PROCEDURE (ETA VALUE STRETCH)
      ENM1=ENO-1
      DO 2196 J=2,ENM1
      ETANEW=ET(J)/CAPK*ETA
      SPX=0.
      SPY=0.
      DUM=RZTRP(SDXETA,XOETA,ETANEW,ENPOLS,SPX)
      XX(I,J)=AIMAG(FZTRP(SDXETA,XOETA,SPX,ENPOLS))
      DUM=RZTRP(SDYETA,YOETA,ETANEW,ENPOLS,SPY)
 2196 YY(I,J)=AIMAG(FZTRP(SDYETA,YOETA,SPY,ENPOLS))
C     END
      IF(XX(I,1).EQ.XX(I,ENO-1).AND.YY(I,1).EQ.YY(I,ENO-1)) WRITE (6,7)I
    7 FORMAT('OGRID LINE NO. ',I4,' HAS DEGENERATED TO POINT.',
     1' TRY THETL,THETT = LE,TE CAMBER ANGLES')
      XX(I,ENO)=X
```

```
2240  YY(I,ENO)=Y                                                            QS036490
C     END                                                                    QS036500
      DO 218 J1=2,ENO                                                        QS036510
      XX(ZN2M1,J1)=XX(1,J1)                                                  QS036520
218   YY(ZN2M1,J1)=YY(1,J1)                                                  QS036530
      RETURN                                                                 QS036540
                                                                             QS036550
                                                                             QS036560
                                                                             QS036570
                                                                             QS036580
                                                                             QS036590
                                                                             QS036600
                                                                             QS036610
                                                                             QS036620
                                                                             QS036630
                                                                             QS036640
                                                                             QS036650
      END                                                                    QS036660
      SUBROUTINE TOSTAG(S,ZET,CHORDO)                                        QS036670
                                                                             QS036680
C      INVERT THE CONFORMAL TRANSFORMATION FROM FLATPLATE CASCADE OF ZERQS036690
C     O STAGGER                                                              QS036700
C      TO THE UNIT CIRCLE. ADD DESIRED SOLIDITY AND STAGGER TERMS AND REQS036710
C     VERT TO                                                                QS036720
C      PHYSICAL PLANE.                                                       QS036730
                                                                             QS036740
C       SOURCE,     HAWTHORNE (LISTED AS REF. 4 OF REF. 2 OF THE USERS MQS036750
C     ANUAL)                                                                 QS036760
                                                                             QS036770
                                                                             QS036780
      INTEGER ZNO,ZN2M1,ENO,ENOPLS                                           QS036790
      DIMENSION ZET(100)                                                     QS036800
      COMMON/GEOM/ZNO,ENO,RLE,RTE,A,CHORD,XUPS,XDNS,STAG,F,CAKK(4),PI,      QS036810
     1LNK,VB,KN,NED,BB,DOSURF,R,SOLID2,ENOPLS                               QS036820
      COMMON /ENTIRE/ XGRID(100,30),YGRID(100,30)                            QS036830
      COMPLEX AI/(0.,1.)/,AHTAN,AHTAN2,AHYPT,ZCMX,WCMX,AQT1,AQT2,READK      QS036840
      COMPLEX ZGRID                                                          QS036850
      READK=1./R                                                             QS036860
      ALAMDA=.5*(READK+1./READK)                                             QS036870
      ZN2M1=2*ZNO-1                                                          QS036880
      N2ZM11=ZN2M1-1                                                         QS036890
      XGRID(1,1)=4.*XGRID(2,1)                                               QS036900
      XGRID(ZN2M1,1)=XGRID(1,1)                                              QS036910
      XGRID(ZNO,1)=-XGRID(1,1)                                               QS036920
      DO  1 I=1,N2ZM11                                                       QS036930
      DO 1 J=1,ENOPLS                                                        QS036940
      SIG=1.                                                                 QS036950
      IF(J.GT.ENO) SIG=-1.                                                   QS036960
```

```
      ZGRID=(XGRID(I,J)+AI*YGRID(I,J)*SIG)
      AHTAN=CEXP((ZGRID        )/2.)-CEXP(-(ZGRID        )/2.)
      AHTAN2=CEXP((ZGRID        )/2.)+CEXP(-(ZGRID        )/2.)
      AHYPT=AHTAN/AHTAN2
      ZCMX= ALAMDA*AHYPT
      WCMX=ZCMX+CSQRT(ZCMX-1.)*CSQRT(ZCMX+1.)
      IF(R.LT.1.1.AND.YGRID(I,J).EQ.0.)WCMX=CMPLX(REAL(WCMX),0.)
      IF(ABS(ABS(REAL(WCMX))-R).LT.1.E-07)
     1AQT1=CMPLX(SIGN(25.,XGRID(I,J)),0.)
      IF(ABS(ABS(REAL(WCMX))-R).GE.1.E-07)
     1AQT1=(CLOG(READK*WCMX+1.)-CLOG(READK*WCMX-1.))
      IF(ZET(I) .LT.0.)AQT1=AQT1-2.*AI*PI

C      ADD STAGGER TERM

      AQT1=AQT1*CEXP(-AI*STAG)
      AQT2=(CLOG(WCMX+READK)-CLOG(WCMX-READK))*CEXP(AI*STAG)

C      CORRECT FOR PROPER SOLIDITY COEFFICIENT AND ASSEMBLE X+IY
      ZCMX=(AQT1+AQT2)/(2.*PI*SOLID2)*CHORD
      XGRID(I,J)=  REAL(ZCMX)+SIN(STAG)/2./SOLID2 *CHORD
      YGRID(I,J)=  SIG*(AIMAG(ZCMX)+COS(STAG)/2./SOLID2 *CHORD)
      YGRID(ZN2M1,J)=YGRID(1,J)
1     XGRID(ZN2M1,J)=XGRID(1,J)
      RETURN
      END
      SUBROUTINE ALTERP(COVERS,STAG,ALFA,R)

C      PERFORMS DOUBLE LINEAR INTERPOLATION ON STAGGER ANGLE AND SOLIDIT
C     Y TO OBTAIN
C      POSITIONS OF VORTEX SINGULARITIES IN THE CONFORMAL PLANE.

C      SOURCE, HAWTHORNE (LISTED AS REF. 4 OF REF. 2. OF USERS MANUAL)



      DIMENSION CSTAB(25,25),GAMTAB(25),RTAB(25),ISTAR(2)

C      TABULATED VALUES FROM THE RANGE OF FIGS B,5C AND B,5D (PP.30,31)
C     OF HAWTHORNE
      DATA RTAB,GAMTAB/1.001,
     1                  1.005,1.01,1.02,1.03,1.04,1.05,1.1,1.2,1.3,1.4
     1,1.5,1.6,1.7,1.8,1.9,2.,2.25,2.5,2.75,3.,3.5,4.,4.5,5.,0.,5.,10.,
     115.,20.,25.,30.,35.,40.,45.,50.,55.,60.,65.,70.,75.,80.,85.,91.,
     195.,100.,4*150./
      PI=3.14159265
      DO 1 J=1,19
      G=GAMTAB(J)*PI/180.
```

```
      DO 1 I=1,25
      RD=RTAB(I)
C     EQNS. FROM FIGS B,5C AND D
      AL=ATAN(TAN(G)*(RD*RD-1.)/(RD*RD+1.))
1     CSTAB(I,J)=1./PI*(COS(G)*ALOG((RD*RD+2.*RD*COS(AL)+1.)/(RD*RD-2.
     1*RD*COS(AL)+1.))+2.*SIN( G)*ATAN(2.*RD*SIN(AL)/(RD*RD-1.)))

      STAGG=STAG*180./PI
      DO 20 J=1,25
      IF(STAGG.LE.GAMTAB(J))GOTO21
20    CONTINUE
      J=25
21    JSTAR=J
      JSM1=J-1
      DO 40 J=JSM1,JSTAR
      J1=J-JSM1+1
      DO 30 I=1,25
      IF(COVERS.GE.CSTAB(I,J))GOTO31
30    CONTINUE
      I=25
31    ISTAR(J1)=I
40    CONTINUE
      ISM1=ISTAR(2)-1
      ISSM1=ISTAR(1)-1
      IS=ISTAR(2)
      ISS=ISTAR(1)
      R1=RTAB(ISS)+(RTAB(ISSM1)-RTAB(ISS))*(COVERS-CSTAB(ISS,JSM1))/
     1(CSTAB(ISSM1,JSM1)-CSTAB(ISS,JSM1))
      R2=RTAB(IS)+(RTAB(ISM1)-RTAB(IS))*(COVERS -CSTAB(IS,JSTAR))/
     1(CSTAB(ISM1,JSTAR)-CSTAB(IS,JSTAR))
      R=R1+(R2-R1)*(STAGG-GAMTAB(JSM1))/(GAMTAB(JSTAR)-GAMTAB(JSM1))
      IF(COVERS.GT.5.)R=1.0001
      ALFA=ATAN(TAN(STAG)*(R*R-1.)/(R*R+1.))
      RETURN
      END
      FUNCTION FUDS(E,Z,SCK)
C     CALCULATES FUNCTION  'T' IN EQN. 1., REF. 2. OF USER MANUAL

      COMMON/GEOM/ DUMP(19),G
      COMMON/FUNCOS/S,C,D,SN,CN,DN
      CALL  JELF(S,C,D,E,SCK)
      CALL  JELF(SN,CN,DN,Z,G)
      B=S*DN*S*DN+(C*D*SN*CN)**2
```

```
      FUDS = SQRT(B)                                                    QS037930
      RETURN                                                            QS037940
      END                                                               QS037950
      FUNCTION OWOW(SCK)                                                QS037960
                                                                        QS037970
C      CALCULATES FUNCTION 'OMEGA' IN EQN. 1., REF. 2 OF USERS MANUAL   QS037980
                                                                        QS037990
                                                                        QS038000
      COMMON/FUNCOS/S,C,D,SN,CN,DN                                      QS038010
      IF(S.EQ.0.)S=1.E-20                                               QS038020
      B= C*D*SN*CN/S/DN                                                 QS038030
      OWOW = ATAN(B)                                                    QS038040
      RETURN                                                            QS038050
      END                                                               QS038060
      FUNCTION BADZAC(SCK)                                              QS038070
                                                                        QS038080
C      CALCULATES FUNCTIUON 'GAMMA' IN EQN. 1., REF. 2 OF USERS MANUAL  QS038090
                                                                        QS038100
                                                                        QS038110
      COMMON/FUNCOS/S,C,D,SN,CN,DN                                      QS038120
      BADZAC = CN*CN+(1.-SCK)*S*S*SN*SN                                 QS038130
      RETURN                                                            QS038140
      END                                                               QS038150
      SUBROUTINE JELF(SN,CN,DN,X,SCK)                                   QS038160
                                                                        QS038170
C      COMPUTES JACOBIAN ELLIPTIC FCNS SN,CN,DN                         QS038180
                                                                        QS038190
C      IF X IS GIVEN AS THE ELLIPTIC INTEGRAL OF THE FIRST KIND WITH MODQS038200
C     ULUS K                                                            QS038210
C      FROM ZERO TO SIN(PHI)  WITH  0<PHI<PI/2, THEN                    QS038220
C              SCK = 1-K*K                                              QS038230
C                 SN(X,K)=SIN(PHI)                                      QS038240
C                 CN(X,K)=COS(PHI)                                      QS038250
C   AND           DN(X,K)=SQRT(1.-K*K*SIN(PHI)**2)                      QS038260
                                                                        QS038270
                                                                        QS038280
                                                                        QS038290
      DIMENSION ARI(12),GEO(12)                                         QS038300
      CM=SCK                                                            QS038310
      Y=X                                                               QS038320
      IF(SCK) 3,1,4                                                     QS038330
1     D=EXP(X)                                                          QS038340
      A=1./D                                                            QS038350
      B=A+D                                                             QS038360
      CN=2./B                                                           QS038370
      DN=CN                                                             QS038380
      SN=TANH(X)                                                        QS038390
2     RETURN                                                            QS038400
```

```
3       D=1.-SCK                                                        QS038410
        CM=-SCK/D                                                       QS038420
        D=SQRT(D)                                                       QS038430
        Y=D*X                                                           QS038440
4       A=1.                                                            QS038450
        DN=1.                                                           QS038460
        DO 6 I=1,12                                                     QS038470
        L=I                                                             QS038480
        ARI(I)=A                                                        QS038490
        CM=SQRT(CM)                                                     QS038500
        GEO(I)=CM                                                       QS038510
        C=(A+CM)*.5                                                     QS038520
        IF(ABS(A-CM)-1.E-4*A)7,7,5                                      QS038530
5       CM=A*CM                                                         QS038540
6       A=C                                                             QS038550
7       Y=C*Y                                                           QS038560
        SN=SIN(Y)                                                       QS038570
        CN=COS(Y)                                                       QS038580
        IF(SN)8,13,8                                                    QS038590
8       A=CN/SN                                                         QS038600
        C=A*C                                                           QS038610
        DO 9 I=1,L                                                      QS038620
        K=L-I+1                                                         QS038630
        B=ARI(K)                                                        QS038640
        A=C*A                                                           QS038650
        C=DN*C                                                          QS038660
        DN=(GEO(K)+A)/(B+A)                                             QS038670
9       A=C/B                                                           QS038680
        A=1./SQRT(C*C+1.)                                               QS038690
        IF(SN)10,11,11                                                  QS038700
10      SN=-A                                                           QS038710
        GOTO 12                                                         QS038720
11      SN=A                                                            QS038730
12      CN=C*SN                                                         QS038740
13      IF(SCK)14,2,2                                                   QS038750
14      A=DN                                                            QS038760
        DN=CN                                                           QS038770
        CN=A                                                            QS038780
        SN=SN/D                                                         QS038790
        RETURN                                                          QS038800
        END                                                             QS038810
        SUBROUTINE XYCALC(KSTART,K2,Z,X,Y)                              QS038820
C               -- XYCALC --                                            QS038830
C.....GENERATES DATA FILES FOR ON-BODY POINTS.                          QS038840
                                                                        QS038850
        INTEGER SGEN                                                    QS038860
        REAL                               C(400),S(400),SP(400)        QS038870
        COMPLEX Z(400),DZ(400),D2Z,ZZ,FZTRP,DZTRP                       QS038880
```

```
      LOGICAL THIN,EVEN,SPGEN,DONE
      COMMON /MAN/ DELSMX,PIO2,DELS1,IHUB
      DIMENSION X(400),Y(400)
      COMMON /SPGENC/       A,DSMAX,RMAX,THIN,B,TMIN,DSEND
      DATA NSMAX,EMPTY/400,1.0E20/

C.....INITIALIZE PROGRAM.
      S1=0.0
      NMAX=400
      DSMAX=DELSMX
      DELS1=DELSMX
      THIN=.FALSE.
      RMAX=1.2
      DSEND=DSMAX
      DONE=.FALSE.
      EVEN=.FALSE.
      B=0.3
      TMIN=0.1
      NFIN=0
      NSP=0

C.....INPUT BODY POINTS AND BODY TYPE.
      DO 25 I=1,NSMAX
      IF (REAL(Z(I)).EQ.EMPTY) GO TO 30
25    NS=I
30    S(1)=S1
      IBAD=SGEN(S,Z,NS)
C     WRITE(6,140)
140   FORMAT(31HOTHE FOLLOWING IS S FROM XYCALC///)
C     WRITE(6,*)S
C     WRITE(6,145)
145   FORMAT(31HOTHE FOLLOWING IS Z FROM XYCALC///)
C     WRITE(6,*)Z
C     WRITE(6,150)
150   FORMAT(32HOTHE FOLLOWING IS NS FROM XYCALC///)
C     WRITE(6,*)NS
      IF (IBAD.NE.0) WRITE (6,125)IBAD

C.....SET UP DERIVATIVES + CURVATURES.
      DO 35 I=1,NS
35    DZ(I)=DZTRP(S,Z,S(I),NS)
      DO 40 I=1,NS
      D2Z=DZTRP(S,DZ,S(I),NS)
40    C(I)=AIMAG(CONJG(DZ(I))*D2Z)/CABS(DZ(I))**3

C.....INPUT AUXILLIARY (CONTROL) DATA.
      IF (NFIN.EQ.0) NFIN=NS
      SFIN=S(NFIN)
```

```
      DSMAX=AMAX1(DSMAX,DSEND)

C.....GENERATE BODY POINTS ON A SEGMENT.
      IF (.NOT.SPGEN(S,Z,C,NS,SP,NSP,SFIN,NMAX)) GO TO 130


C.....OUTPUT RESULTING ON-BODY POINTS.
85    DO 90 I=1,NSP
      IBY=KSTART+I-1
      ZZ=FZTRP(S,Z,SP(I),NS)
C     DZ(I)=DZTRP(S,Z,SP(I),NS)
C     D2Z=DZTRP(S,DZ,SP(I),NS)
C     C(I)= AIMAG(CONJG(DZ(I))*D2Z)/CABS(DZ(I))**3
      X(IBY)=REAL(ZZ)
90    Y(IBY)=AIMAG(ZZ)
      K1=KSTART
      K2=NSP+KSTART-1
120   RETURN
C.....ERROR MESSAGES.
125   FORMAT(21H SGEN FAILED.  IBAD= ,I3)
130   WRITE (6,135)
135   FORMAT(34H SPGEN UNABLE TO COMPLETE SEGMENT )
      STOP
      END
      INTEGER FUNCTION SGEN(S,F,NS)
      REAL S(400)
      COMPLEX F(400),DZTRP
      DATA MAX,N,FN,TEST/4,10,10.0,0.01/
      DO 10 I=2,NS
10    S(I)=S(I-1)+CABS(F(I)-F(I-1))
      DO 30 K=1,MAX
      SGEN=0
      DO 25 I=2,NS
      DS=S(I)-S(I-1)
      DARG=DS/FN
      ARGO=S(I-1)-DARG/2.0
      SUM=0.0
      DO 15 J=1,N
      ARG=ARGO+FLOAT(J)*DARG
      TVQQ=CABS(DZTRP(S,F,ARG,NS))-1.0
15    SUM=SUM+TVQQ
      SUM=SUM/FN
      ERROR=ABS(SUM)
C     WRITE(6,19)
19    FORMAT(42H0THE FOLLOWING IS TVQQ,SUM,ERROR FROM SGEN)
C     WRITE(6,*)TVQQ,SUM,ERROR
      DS=DS*SUM
      DO 20 J=1,NS
```

```
20    S(J)=S(J)+DS
      IF(ERROR.GT.TEST.AND.SGEN.EQ.0)SGEN=I
25    CONTINUE
      IF(SGEN.EQ.0)RETURN
30    CONTINUE
      RETURN
      END
      LOGICAL FUNCTION SPGEN (S,Z,C,NS,SP,NSP,SFIN,NMAX)
C.....GENERATES TABLE SP HAVING VALUES OF PARAMETER S AS WIDELY SPACED
C     AS POSSIBLE AND YET SATISFYING THE FOLLOWING CONDITIONS ON DS:
C         1    NSP  .LE. NMAX
C         2    DS   .LE. A/C(S)      (C=CURVATURE)
C         3    DS   .LE. DSMAX
C         4A   DS(I) .LE. DS(I-1)*RMAX
C         4B   DS(I) .GE. DS(I-1)/RMAX
C     FOR THIN SECTIONS, AN ADDITIONAL CONDITION IS:
C              DS   .LE. B*TLOC      (TLOC=LOCAL THICHNESS)
C.....SPGEN = .TRUE. IF ALL CONDITIONS HAVE BEEN SATISFIED.

      REAL S(400),C(400),SP(400)
      COMPLEX Z(400),FZTRP
      LOGICAL THIN,FIN
      COMMON /SPGENC/      A,DSMAX,RMAX,THIN,B,TMIN,DSEND
      COMMON /MAN/ DELSMX,PIO2,DELS1,IHUB
      DATA ONE,CMIN/1.0001,1.0E-6/

C.....INITIALIZATION SECTION.
      SPGEN=.FALSE.
      J1=MAXO(NSP,2)+1
      IF (NSP.GT.1) GO TO 15
      IF (NSP.LT.1) SP(1)=S(1)
      DS1=    DELS1
10    SP(2)=SP(1)+DS1
C     WRITE(6,51)
51    FORMAT(30HOTHE FOLLOWING IS S FROM SPGEN//////)
C     WRITE(6,*)S
C     WRITE(6,52)
52    FORMAT(30HOTHE FOLLOWING IS C FROM SPGEN///////)
C     WRITE(6,*)C
C     WRITE(6,53)
53    FORMAT(33HOTHE FOLLOWING IS SBAR FROM SPGEN//////)
C     WRITE(6,*)SBAR
C     WRITE(6,54)
54    FORMAT(31HOTHE FOLLOWING IS NS FROM SPGEN////////)
C     WRITE(6,*)NS
C.....BEGIN MAIN LOOP.
15    DO 45 J=J1,NMAX
      L=J
```

```
20    I=L
25    DSLAST=SP(I-1)-SP(I-2)
      SBAR=SP(I-1)+DSLAST/2.0
      CA=AMAX1(CMIN,ABS(FNTRP(S,C,SBAR,NS)))
      DSLIM=AMIN1(DS1   ,DSLAST*RMAX)
      IF (.NOT.THIN) GO TO 30
      TLOC=CABS(FZTRP(S,Z,SBAR,NS)-FZTRP(S,Z,S(NS)-SBAR,NS))
      DSLIM=AMIN1(DSLIM,B*AMAX1(TLOC,TMIN))
30    DSFIN=SFIN-SP(I-1)
      NEVEN=DSFIN/DSLIM/ONE+1.0
      DSEVEN=DSFIN/FLOAT(NEVEN)
      DS=AMIN1(A/CA,DSEVEN)
      IF (I.NE.J) DS=AMIN1(DS,DSLAST/RMAX)

C.....CALCULATED VALUE OF DS SATISFIES CONDITIONS 2 THRU 4A.TEST FOR 4B.
      IF (DS.GE.DSLAST/RMAX) GO TO 40

C.....IF CONDITION 4B IS NOT SATISFIED, RE-DO EARLIER INTERVALS
C.....USING SMALLER VALUES OF DS.  IF RE-DOING ALL INTERVALS WON'T
C.....WORK, START OVER USING SMALLER STARTING VALUE OF DS (DS1).
35    L=L-1
      IF (L.GE.J1) GO TO 20
      IF (NSP.GT.1) RETURN
      DS1=DS1/RMAX
      GO TO 10

C.....IF CONDITIONS 2 THRU 4B ARE SATISFIED, TEST FOR FINISH.
40    SP(I)=SP(I-1)+DS
      FIN=SFIN/SP(I).LE.ONE
      IF (FIN.AND.DS.GT.DSEND) GO TO 35
      IF (FIN) GO TO 50
      IF (I.GE.J) GO TO 45
      I=I+1
      GO TO 25
45    CONTINUE
C.....SPGEN =.FALSE. IF CONDITION 1 CANNOT BE SATISFIED.
      RETURN

C.....IF CONDITIONS ARE SATISFIED, UPDATE NSP.
50    NSP=I
      DELSI=DS
      SPGEN=.TRUE.
      RETURN
      END
      COMPLEX FUNCTION DZTRP (A,F,X,NA)
C.....COMPLEX DERIVATIVE EVALUATION FOR DOUBLE 3-POINT INTERPOLATION.
      COMPLEX F(400)
      REAL A(400)
```

```
      COMMON /NTRPC3/ I1,I2,C(4)
C.....FIRST EVALUATE FUNCTION COEFFICIENTS.
      CALL FNTRPA (A,X,NA)
      CALL DNTRPC
C.....THEN EVALUATE FUNCTION VALUE.
C     GO TO 10
C     ENTRY DZTRP1 (F)
C     DZTRP1=0.
10    DZTRP=0.0
      J=0
      DO 1 I=I1,I2
      J=J+1
1     DZTRP=DZTRP+C(J)*F(I)
      RETURN
      END
      COMPLEX FUNCTION FZTRP (A,F,X,NA)
C.....COMPLEX FUNCTION EVALUATION BY DOUBLE 3-POINT INTERPOLATION.
      COMPLEX F(400)
      REAL A(400)
      COMMON /NTRPC2/ I1,I2,C(4)
C.....FIRST EVALUATE FUNCTION COEFFICIENTS.
      CALL FNTRPA (A,X,NA)
      CALL FNTRPC
C.....THEN EVALUATE FUNCTION VALUE.
C     GO TO 10
C     ENTRY FZTRP1 (F)
C     FZTRP1=0.
10    FZTRP=0.0
      J=0
      DO 1 I=I1,I2
      J=J+1
1     FZTRP=FZTRP+C(J)*F(I)
      RETURN
      END
      FUNCTION FNTRP (A,F,X,NA)
C.....FUNCTION EVALUATION FOR DOUBLE 3-POINT INTERPOLATION.
      REAL F(400),A(400)
      COMMON /NTRPC2/ I1,I2,C(4)
C.....FIRST EVALUATE FUNCTION COEFFICIENTS.
      CALL FNTRPA (A,X,NA)
      CALL FNTRPC
C.....THEN EVALUATE FUNCTION VALUE.
C     GO TO 5
C     ENTRY FNTRP1 (F)
C     FNTRP1=0.
5     FNTRP=0.0
      J=0
      DO 10 I=I1,I2
```

```
      J=J+1                                                             QS041290
10    FNTRP=FNTRP+C(J)*F(I)                                             QS041300
      RETURN                                                            QS041310
      END                                                               QS041320
      SUBROUTINE DNTRPC                                                 QS041330
C.....CALCULATION OF C COEFFICIENTS FOR DERIVATIVES OF DOUBLE           QS041340
C.....3-POINT INTERPOLATION.                                            QS041350
      COMMON /NTRPC1/ L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44       QS041360
      COMMON /NTRPC3/ I1,I2,C1,C2,C3,C4                                 QS041370
      IF (L.LE.1) GO TO 13                                              QS041380
      IF (L-3) 12,11,10                                                 QS041390
C.....FOR DOUBLE 3-POINT INTERPOLATION.                                 QS041400
10    C1=+(A22+A33+A22)/A23*A33/A12/A13                                 QS041410
      C4=-(A33+A22+A33)/A23*A22/A34/A24                                 QS041420
      P=A23*A23                                                         QS041430
      C2=-((A11+A33+A11)*A33/A12+(A33*A44+A22*A44+A22*A33)/A24)/P       QS041440
      C3=+((A44+A22+A44)*A22/A34+(A22*A11+A33*A11+A33*A22)/A13)/P       QS041450
      GO TO 14                                                          QS041460
C.....FOR SIMPLE 3-POINT INTERPOLATION.                                 QS041470
11    C1=+(A33+A22)/A12/A13                                             QS041480
      C2=-(A33+A11)/A12/A23                                             QS041490
      C3=+(A22+A11)/A13/A23                                             QS041500
      GO TO 14                                                          QS041510
C.....FOR 2-POINT INTERPOLATION.                                        QS041520
12    C1=1.0/A12                                                        QS041530
      C2=-C1                                                            QS041540
      GO TO 14                                                          QS041550
C.....ONLY ONE TABLE VALUE GIVEN.                                       QS041560
13    C1=0.0                                                            QS041570
14    I1=I                                                              QS041580
      I2=I+L-1                                                          QS041590
      RETURN                                                            QS041600
      END                                                               QS041610
      SUBROUTINE FNTRPA(A,X,NA)                                         QS041620
C.....COMMON SUBROUTINE EVALUATES A COEFFICIENTS IN DOUBLE              QS041630
C.....3-POINT INTERPOLATIONS.                                           QS041640
C       L=NO. OF POINTS IN THE FIT                                      QS041650
C       I=INDEX TO FIRST POINT                                          QS041660
      REAL A(400)                                                       QS041670
      COMMON /NTRPC1/ L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44       QS041680
C.....GET I AND L BY TABLE LOOK-UP.                                     QS041690
      L=LIMIT (1,NA,3)                                                  QS041700
      M=MAXO(1,NA-2)                                                    QS041710
      CALL TLU (A,X,NA,J)                                               QS041720
      IF (J.EQ.LIMIT(2,J,M)) L=4                                        QS041730
      I=LIMIT(1,J-1,M)                                                  QS041740
C.....CALCULATE A-ARRAY.                                                QS041750
      A11=A(I)                                                          QS041760
```

```
      A22=A(I+1)                                                        QS041770
      A33=A(I+2)                                                        QS041780
      IF (L.NE.4) IF (L-2) 20,15,10                                     QS041790
      A44=A(I+3)                                                        QS041800
      A14=A11-A44                                                       QS041810
      A24=A22-A44                                                       QS041820
      A34=A33-A44                                                       QS041830
      A44=X-A44                                                         QS041840
10    A13=A11-A33                                                       QS041850
      A23=A22-A33                                                       QS041860
      A33=X-A33                                                         QS041870
15    A12=A11-A22                                                       QS041880
      A22=X-A22                                                         QS041890
      A11=X-A11                                                         QS041900
20    RETURN                                                            QS041910
      END                                                               QS041920
      SUBROUTINE FNTRPC                                                 QS041930
C.....CALCULATION OF C COEFFICIENTS FOR FUNCTION VALUES BY DOUBLE       QS041940
C.....3-POINT INTERPOLATION.                                            QS041950
      COMMON /NTRPC1/L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44        QS041960
      COMMON /NTRPC2/ I1,I2,C1,C2,C3,C4                                 QS041970
      IF (L.LE.1) GO TO 25                                              QS041980
      IF (L-3) 20,15,10                                                 QS041990
C.....FOR DOUBLE 3-POINT INTERPOLATION.                                 QS042000
10    C1=+A33/A23*A22/A12*A33/A13                                       QS042010
      C4=-A22/A23*A33/A34*A22/A24                                       QS042020
      P2=A33/A23*A11/A23                                                QS042030
      P3=A22/A23*A44/A23                                                QS042040
      C2=-A33*(P2/A12+P3/A24)                                           QS042050
      C3=+A22*(P3/A34+P2/A13)                                           QS042060
      GO TO 30                                                          QS042070
C.....FOR SIMPLE 3-POINT INTERPOLATION.                                 QS042080
15    C1=+A22/A12*A33/A13                                               QS042090
      C2=-A11/A12*A33/A23                                               QS042100
      C3=+A11/A13*A22/A23                                               QS042110
      GO TO 30                                                          QS042120
C.....FOR 2-POINT INTERPOLATION.                                        QS042130
20    C1=+A22/A12                                                       QS042140
      C2=-A11/A12                                                       QS042150
      GO TO 30                                                          QS042160
C.....ONLY ONE TABLE VALUE GIVEN.                                       QS042170
25    C1=1.0                                                            QS042180
30    I1=I                                                              QS042190
      I2=I+L-1                                                          QS042200
      RETURN                                                            QS042210
      END                                                               QS042220
      SUBROUTINE TLU (TABLE,ARG,N,I)                                    QS042230
C.....TABLE LOOK UP:  FINDS I SUCH THAT                                 QS042240
```

```
C      ARG.GE.TABLE(I).AND.ARG.LT.TABLE(I+1)                                  QS042250
C        IF I=0, ARG.LT.TABLE(1)                                              QS042260
C        IF I=N, ARG.GE.TABLE(N)                                              QS042270
       REAL TABLE(400)                                                        QS042280
       I=LIMIT(1,I,N)                                                         QS042290
       IF (ARG.GE.TABLE(I)) GO TO 15                                          QS042300
C.....DESCEND IN TABLE.                                                       QS042310
10     I=I-1                                                                  QS042320
       IF (I.LE.0) RETURN                                                     QS042330
       IF (ARG.GE.TABLE(I)) RETURN                                            QS042340
       GO TO 10                                                               QS042350
C.....ASCEND IN TABLE.                                                        QS042360
15     IF (I.GE.N) RETURN                                                     QS042370
       IF (ARG.LT.TABLE(I+1)) RETURN                                          QS042380
       I=I+1                                                                  QS042390
       GO TO 15                                                               QS042400
       END                                                                    QS042410
       FUNCTION LIMIT (I,J,K)                                                 QS042420
C.....INTEGER FUNCTION LIMITS J BETWEEN I AND K.                              QS042430
       LIMIT=I                                                                QS042440
       IF (J.LT.LIMIT) RETURN                                                 QS042450
       LIMIT=K                                                                QS042460
       IF (J.GT.LIMIT) RETURN                                                 QS042470
       LIMIT=J                                                                QS042480
       RETURN                                                                 QS042490
       END                                                                    QS042500
       LOGICAL FUNCTION XNTRP (A,F,F0,NA,X)                                   QS042510
C.....INVERSE FUNCTION FOR DOUBLE 3-POINT INTERPOLATION.                      QS042520
                                                                              QS042530
C      GIVEN TABLES A AND F (EACH OF LENGTH NA), THE VALUE OF X IS            QS042540
C      FOUND FOR WHICH FNTRP (A,F,X,NA)=F0.  ONLY SOLUTIONS LARGER            QS042550
C      THAN THE ENTRY VALUE OF X ARE CONSIDERED; ONLY THE SMALLEST            QS042560
C      OF THESE IS RETURNED.  IF NO SOLUTION IS FOUND, XNTRP=.FALSE.          QS042570
C      AND X IS LEFT UNALTERED.                                               QS042580
                                                                              QS042590
       REAL A(400),F(400),C(4),R(3)                                           QS042600
       INTEGER II(3),REALX,RCMPLX,ICMPLX,TYPE                                 QS042610
       LOGICAL LAST,SIZE                                                      QS042620
       COMMON /NTRPC1/ L,I,EXTRA(10)                                          QS042630
       COMMON /NTRPC5/ M,B(4,4)                                               QS042640
       DATA REALX,RCMPLX,ICMPLX,EPSLON/0,1,2,1.0E-5/                          QS042650
       COMPLEX FZ(400)                                                        QS042660
       TYPE=REALX                                                             QS042670
       GO TO 100                                                              QS042680
                                                                              QS042690
C.....ALTERNATE ENTRIES FOR COMPLEX TYPE FUNCTION TABLES:                     QS042700
                                                                              QS042710
C.....RZTRP IS USED TO GET X FOR WHICH REAL(FZTRP(A,F,X,NA))=F0.              QS042720
```

```
      ENTRY RZTRP (A,FZ,FO,NA,X)
      RZTRP=0.
C.... COMPLEX FZ(NA) THIS WAS MOVED UP TO COMPILE PROPERLY ON NPS 370
      TYPE=RCMPLX
      GO TO 100

C.....IZTRP IS USED TO GET X FOR WHICH AIMAG(FZTRP(A,F,X,NA))=FO.
      ENTRY IZTRP (A,FZ,FO,NA,X)
      IZTRP=0.
      TYPE=ICMPLX

C.....INITIALIZATION.
100   XIN=X
      XMAX=XIN
      DX=0.0

C.....CALCULATE POLYNOMIAL COEFFICIENTS.
1     XMIN=AMAX1(XIN+DX,XMAX-DX)
      CALL BNTRP (A,XMAX,NA)
      LAST=M.GE.NA-1
      IF (LAST) GO TO 2
      MAX=M+L-3
      XMAX=A(MAX)
2     DO 3 J=1,L
      C(J)=0.0
      DO 3 N=1,L
      IN=I+N-1
      IF (TYPE.EQ.REALX) C(J)=C(J)+B(N,J)*F(IN)
      IF (TYPE.EQ.RCMPLX) C(J)=C(J)+B(N,J)*REAL(FZ(IN))
      IF (TYPE.EQ.ICMPLX) C(J)=C(J)+B(N,J)*AIMAG(FZ(IN))
3     CONTINUE
      C(1)=C(1)-FO
      IF(L.EQ.4.AND.ABS(C(L)).LT.1.E-04)L=3
      LL=L
4     IF (LL.LT.2) GO TO 8
      IF (C(LL).NE.0.0) GO TO 5
      LL=LL-1
      GO TO 4

C.....GET ROOTS OF POLYNOMIAL.
5     NR=1
      IF (LL.EQ.2) R(1)=-C(1)/C(2)
      IF (LL.EQ.3) NR=NRRTS2(C,R)
      IF (LL.EQ.4) NR=NRRTS3(C,R)

C.....TEST ROOTS OF POLYNOMIAL.
      IF (NR.LE.0) GO TO 8
      IF (.NOT.SIZE(R,II,NR)) CALL ORDER (R,II,NR)
```

```
      DO 6 N=1,NR                                                       QS043210
      XVAL=R(N)+A(M)                                                    QS043220
      IF (XVAL.GT.XMIN) GO TO 7                                         QS043230
6     CONTINUE                                                          QS043240
      GO TO 8                                                           QS043250
7     IF (LAST .OR. XVAL.LE.XMAX) X=XVAL                                QS043260
      IF(ABS(XVAL-XMAX).LE.1.E-04)X=XVAL                                QS043270
8     XNTRP=X.NE.XIN                                                    QS043280
                                                                        QS043290
C.....IF NO SOLUTION FOUND, KEEP SEARCHING.                              QS043300
9     IF (XNTRP.OR.LAST) RETURN                                         QS043310
      DX=EPSLON*(ABS(XMAX)+ABS(XMIN))                                   QS043320
      GO TO 1                                                           QS043330
      END                                                               QS043340
      SUBROUTINE BNTRP (A,X,NA)                                         QS043350
C.....CALCULATION OF MATRIX B BY WHICH COEFFICIENTS OF THE POLYNOMIAL    QS043360
C     USED IN DOUBLE 3-POINT INTERPOLATION MAY BE OBTAINED.             QS043370
C                                                                       QS043380
C     IF      F(X) = C(1) + C(2)*(X-A(M)) + C(3)*(X-A(M))**2 +...       QS043390
C                                                                       QS043400
C     IS THE POLYNOMIAL USED IN THE NEIGHBORHOOD OF X, THE COEFFICIENTS QS043410
C     CAN BE OBTAINED FROM MATRIX B BY:                                 QS043420
C                                                                       QS043430
C             C(J) = B(1,J)*F(I) + B(2,J)*F(I+1) +...                   QS043440
C                                                                       QS043450
C     FOR A TOTAL OF L VALUES OF C AND F (AT MOST 4) WHERE L AND I ARE  QS043460
C     FOUND IN COMMON /NTRPC1/ AND M AND B ARE FOUND IN COMMON /NTRPC5/.QS043470
                                                                        QS043480
      REAL A(NA)                                                        QS043490
      COMMON /NTRPC1/ L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44       QS043500
      COMMON /NTRPC5/ M,B11,B21,B31,B41,B12,B22,B32,B42,B13,B23,B33,    QS043510
     1                B43,B14,B24,B34,B44                               QS043520
                                                                        QS043530
C.....SET UP COMMON BLOCK /NTRPC1/                                       QS043540
      CALL FNTRPA (A,X,NA)                                              QS043550
      IF (L.GT.1) GO TO 2                                               QS043560
                                                                        QS043570
C.....ENTER HERE FOR ONLY ONE COEFFICIENT.                               QS043580
1     M=I                                                               QS043590
      B11=1.0                                                           QS043600
      RETURN                                                            QS043610
                                                                        QS043620
C.....ENTER HERE FOR 2 OR MORE COEFFICIENTS.                             QS043630
2     M=I+1                                                             QS043640
      Q=1.0/A12                                                         QS043650
      IF (L.GT.2) GO TO 3                                               QS043660
      B12=Q                                                             QS043670
      B22=-Q                                                            QS043680
```

```
      GO TO 24
C.....ENTER HERE FOR 3 OR MORE COEFFICIENTS.
3     P=1.0/A23
      IF (L.GT.3) GO TO 4
      B13=Q/A13
      B23=-Q*P
      B33=P/A13
      GO TO 34
C.....ENTER HERE FOR 4 COEFFICIENTS.
4     P2=P*P
      B14=P*Q/A13
      B24=-P2/A12-P2/A24
      B34=P2/A34+P2/A13
      B44=-P/A34/A24
      B13=(Q+Q)/A13
      B23=-P*(Q+Q)-P/A24
      B33=P2*(A24/A34+(A23-A12)/A13)
      B43=-1.0/A34/A24
      B41=0.0
      B42=0.0
34    B12=A23/A12/A13
      B22=P-Q
      B32=-A12/A13/A23
      B31=0.0
24    B11=0.0
      B21=1.0
      RETURN
      END
      INTEGER FUNCTION NRRTS3 (C,R)
C.....FINDS REAL ROOTS R, OF A CUBIC WITH COEFFICIENTS C, WHEN:
C              C(1) + C(2)*X + C(3)*X**2 + C(4)*X**3 = 0.0

      REAL C(4),R(3),K1,K2
      DATA THIRD,K1,K2/0.3333333,2.094395,4.188790/
      QRT(ARG)=SIGN(ABS(ARG)**THIRD,ARG)

C.....CONVERT TO NORMAL FORM AND CALCULATE NORMAL COEFFICIENTS.
      CNORM=3.0*C(4)
      P=C(3)/CNORM
      Q=C(2)/CNORM
      B=P**3-1.5*(P*Q-C(1)/CNORM)
      B2=B**2
      A=Q-P*P
      A3=A**3
      RAD=B2+A3
      IF (RAD) 3,2,1
```

```
C.....THERE IS ONE REAL ROOT.
1     NRRTS3=1
      IF (RAD.NE.0.0) RAD=SQRT(RAD)
      A=-B+RAD
      B=-B-RAD
      IF (A.NE.0.0) A=QRT(A)
      IF (B.NE.0.0) B=QRT(B)
      R(1)=A+B
      GO TO 10

C.....THERE ARE TWO REAL ROOTS.
2     NRRTS3=2
      IF (B.EQ.0.0) GO TO 1
      R(1)=QRT(B)
      R(2)=-2.0*R(1)
      GO TO 10

C.....THERE ARE THREE REAL ROOTS.
3     NRRTS3=3
      PHI= ARCOS(SIGN(SQRT(-B2/A3),-B))/3.0
      CR=2.0*SQRT(-A)
      R(1)=CR*COS(PHI)
      R(2)=CR*COS(PHI+K1)
      R(3)=CR*COS(PHI+K2)

C.....CONVERT BACK TO ORIGINAL FORM.
10    DO 11 I=1,NRRTS3
11    R(I)=R(I)-P
      RETURN

      ENTRY NRRTS2 (C,R)
C.....FINDS REAL ROOTS R, OF A QUADRATIC WITH COEFFICIENTS C, WHEN:
C              C(1) + C(2)*X + C(3)*X**2 + = 0.0


C.....CALCULATE RADICAL.
      RAD=C(2)**2-4.0*C(3)*C(1)
      IF (RAD) 21,22,23

C.....NO REAL ROOTS.
21    NRRTS3=0
      NRRTS2=0
      RETURN
C.....ONE REAL ROOT.
22    NRRTS3=1
      R(1)=-C(2)/2.0/C(3)
      NRRTS2=1
```

```
      RETURN
C.....TWO REAL ROOTS.
23    NRRTS3=2
      RAD=SQRT(RAD)
      TWOA=2.0*C(3)
      R(1)=-(RAD+C(2))/TWOA
      R(2)=+(RAD-C(2))/TWOA
      NRRTS2=2
      RETURN
      END
      LOGICAL FUNCTION SIZE (X,I,N)
C.....CREATES THE POINTER ARRAY I, ORDERED BY SIZE OF X.
C.....VALUE IS .TRUE. IF X ARRAY IS IN ORDER.
      DIMENSION X(3),I(3)
      COMPLEX Z(3),ZS
      SIZE=.TRUE.
      DO 2 J=1,N
      K=J
1     IF (K.LE.1) GO TO 2
      L=I(K-1)
      IF (X(J).GE.X(L)) GO TO 2
      I(K)=L
      K=K-1
      SIZE=.FALSE.
      GO TO 1
2     I(K)=J
      RETURN
      ENTRY ORDER (X,I,N)
      ORDER=0.
C.....REARRANGES X ARRAY ACCORDING TO POINTER ARRAY I.
      DO 14 J=1,N
      IF (I(J).LE.0) GO TO 13
      M=J
      S=X(M)
11    L=I(M)
      I(M)=-L
      IF (L.EQ.J) GO TO 12
      X(M)=X(L)
      M=L
      GO TO 11
12    X(M)=S
13    I(J)=IABS(I(J))
14    CONTINUE
      RETURN
      ENTRY ZORDER (Z,I,N)
      ZORDER=0.
C.....REARRANGES Z (COMPLEX) ARRAY ACCORDING TO POINTER ARRAY I.
```

```
C     COMPLEX Z(N),ZS THIS WAS MOVED UP TO COMPILE PROPERLY ON NPS 370
      DO 24 J=1,N
      IF (I(J).LE.0) GO TO 23
      M=J
      ZS=Z(M)
21    L=I(M)
      I(M)=-L
      IF (L.EQ.J) GO TO 22
      Z(M)=Z(L)
      M=L
      GO TO 21
22    Z(M)=ZS
23    I(J)=IABS(I(J))
24    CONTINUE
      RETURN
      END
      SUBROUTINE PJGRID(ZET,ET,ZCAM,IMAXY,PITCH)

C           ELECTROSTATIC ANALOG GRID GENERATOR

C                  J. J. ADAMCZYK
C           NASA LEWIS RESEARCH CENTER 1980

C               (SEE REF. 2, QSONIC USERS MANUAL)

      COMMON/ENTIRE/  X(100,30),Y(100,30),ETA(100,30),ZETA(100,30)
      COMMON/CROSS/XQ(4,100),YQ(4,100),EQ(4,100),ZQ(4,100),ZETB(100),
     1STREN(63),FUN(100),GUN1(100),ARG(100),ZBODY(63),DTREN(63)
      COMMON/GEOM/NZGRID,NEGRID,RLE,RTE,C,CHORB,XUPS,XDNS,STAG,CC,CAP(4)
     1,PI
     1,RNK,VB,KN,NUM
      COMPLEX A,B,ZO,ZA,ZB,ZTEMP,ZTEMQ
      COMPLEX SUM3,SUM4,SUM5,SUM6,STREN
      COMPLEX SUMD,HSIN
      COMPLEX ZTEL,AI,HTAN,Z,SUM
      COMPLEX SOUR,ZCAM,ZBODY,ZPET
      COMPLEX ZST
      DOUBLE PRECISION DTREN,BINF
      DIMENSION ZPET(100,30),ZCAM(100),ET(100),ZET(100)
      EQUIVALENCE (ZPET(1,1),ETA(1,1))
      COMMON/CRHOSS/ BINF(63,63),SOUR(63,63)
      LOGICAL XNTRP,TEST
      NETAO=50
      NZETO=50
      NZET4=50
```

```
      IMID=1                                                            QS045610
      NUMX=31                                                           QS045620
      NUMY=14                                                           QS045630
      NUMZ=16                                                           QS045640
C                                                                       QS045650
C     NZGRID MUST BE ODD                                                QS045660
C                                                                       QS045670
      AI=CMPLX(0.0,1.0)                                                 QS045680
      PI=3.1415927                                                      QS045690
      CHORD=CHORB*COS(STAG)                                             QS045700
      STAG=STAG*180./PI                                                 QS045710
      S=PITCH                                                           QS045720
      WRITE(6,1006) NEGRID,NZGRID,IMAXY                                 QS045730
1006  FORMAT(5X,'NEGRID= ',I3,' NZGRID,ODD= ',I3,' IMAXY= ',I3)         QS045740
10    CONTINUE                                                          QS045750
      WRITE(6,1020)                                                     QS045760
1020  FORMAT(10X,'SURFACE COORDINATES')                                 QS045770
      WRITE(6,1030) (ZCAM(I),I=1,NUM)                                   QS045780
1030  FORMAT(7X,2F12.7)                                                 QS045790
      WRITE(6,1040) NUM,PITCH,CHORD,STAG                                QS045800
1040  FORMAT(4X,'NUMBER OF BODY POINTS',I5,10X,'PITCH',F12.7,10X,' CHO 'QS045810
     1,F12.7,' STAGGER',F12.7)                                          QS045820
      ZST=ZCAM(1)                                                       QS045830
      DO 25 I=1,NUM                                                     QS045840
      ZCAM(I)=ZCAM(I)-ZST                                               QS045850
25    CONTINUE                                                          QS045860
      MUM=NUM-1                                                         QS045870
      PSAVE=PITCH                                                       QS045880
      PITCH=2.0*PITCH                                                   QS045890
      XMAX=0.0                                                          QS045900
      IMAX=1                                                            QS045910
      DO 6000 I=1,NUM                                                   QS045920
      IF(XMAX .LE. REAL(ZCAM(I))) XMAX=REAL(ZCAM(I))                    QS045930
      IF(XMAX .LE. REAL(ZCAM(I))) IMAX=I                                QS045940
6000  CONTINUE                                                          QS045950
      DO 30 I=1,MUM                                                     QS045960
      IF(IMID .EQ. 1) ZBODY(I)=(ZCAM(I)+ZCAM(I+1))/2.0                  QS045970
      DO 30 J=1,MUM                                                     QS045980
      A=ZCAM(J)                                                         QS045990
      B=ZCAM(J+1)                                                       QS046000
      Z0=(A+B)/2.0                                                      QS046010
      ZA=CLOG(PI/PITCH*(ZBODY(I)-A))                                    QS046020
      ZB=CLOG(PI/PITCH*(ZBODY(I)-B))                                    QS046030
      IF(I .EQ. J) ZA=ZA                                                QS046040
      IF(I .EQ. J  .AND. AIMAG(ZB) .LT. 0.0 .AND. J .LT. IMAX) ZB=ZB+2.0QS046050
     **AI*PI                                                            QS046060
      IF(I .LT. J .AND. AIMAG(ZA) .LT. 0.0 .AND. I .LT. IMAXY) ZA=ZA+2.0QS046070
     **AI*PI                                                            QS046080
```

```
      IF(I .LT. J .AND. AIMAG(ZB) .LT. 0.0 .AND. I .LT. IMAXY) ZB=ZB+2.0QS046090
     **AI*PI                                                           QS046100
      IF(I .GT. J) ZA=ZA                                               QS046110
      IF(I .GT. J) ZB=ZB                                               QS046120
      IF(I .GE. J .AND. J .GE. IMAX/2.0 .AND. AIMAG(ZA) .GT. PI/2.0) ZA=QS046130
     *ZA-2.0*PI*AI                                                     QS046140
      IF(I .GT. J .AND. J.GE. IMAX/2.0 .AND. AIMAG(ZB) .GT. PI/2.0)ZB=ZBQS046150
     *-2.0*PI*AI                                                       QS046160
 4000 CONTINUE                                                         QS046170
      SOUR(I,J)=(ZBODY(I)-A)*ZA-(ZBODY(I)-B)*ZB+(A-B)                  QS046180
      ZTEL=(ZBODY(I)-ZO)*PI/PITCH                                      QS046190
      HTAN=(CEXP(ZTEL)-CEXP(-ZTEL))/(CEXP(ZTEL)+CEXP(-ZTEL))           QS046200
      IF(CABS(ZTEL) .EQ. 0.0) SOUR(I,J)=SOUR(I,J)                      QS046210
      IF(CABS(ZTEL) .NE. 0.0) SOUR(I,J)=SOUR(I,J)+CLOG(HTAN/ZTEL)*(B-A) QS046220
      A1=ABS(REAL(B-A))                                                QS046230
      A2=ABS(AIMAG(B-A))                                               QS046240
      IF(A1 .GE. A2) BINF(I,J)=REAL(SOUR(I,J))+AIMAG(B-A)/REAL(B-A)*AIMAQS046250
     *G(SOUR(I,J))                                                     QS046260
      IF(A1 .LT. A2) BINF(I,J)=-AIMAG(SOUR(I,J))-REAL(B-A)/AIMAG(B-A)*REQS046270
     *AL(SOUR(I,J))                                                    QS046280
   30 CONTINUE                                                         QS046290
C                                                                      QS046300
C     SOLUTION OF LINEAR EQUATIONS FOR ELECTROSTATIC CHARGE STRENGTH DISQS046310
C     TRIBUTION                                                        QS046320
C                                                                      QS046330
      IF(NUM .GT. 1) GO TO 40                                          QS046340
      STREN(1)=1.0/BINF(1,1)                                           QS046350
      GO TO 50                                                         QS046360
   40 CONTINUE                                                         QS046370
      DO 45 I=1,MUM                                                    QS046380
      DTREN(I)=1.0D0                                                   QS046390
   45 CONTINUE                                                         QS046400
                                                                       QS046410
                                                                       QS046420
      CALL CHLSKY(BINF,DTREN,MUM)                                      QS046430
   50 CONTINUE                                                         QS046440
                                                                       QS046450
                                                                       QS046460
                                                                       QS046470
      DO 51 I=1,MUM                                                    QS046480
      A=ZCAM(I)                                                        QS046490
      B=ZCAM(I+1)                                                      QS046500
      A1=ABS(REAL(B-A))                                                QS046510
      A2=ABS(AIMAG(B-A))                                               QS046520
      STEMP=DTREN(I)                                                   QS046530
      IF(A1 .GE. A2)  STREN(I)=STEMP-AI*STEMP*AIMAG(B-A)/REAL(B-A)     QS046540
      IF(A1 .LT. A2) STREN(I)=-STEMP*REAL(B-A)/AIMAG(B-A)+AI*STEMP     QS046550
   51 CONTINUE                                                         QS046560
```

```
      DO 55 I=1,MUM                                                     QS046570
      SUM2=0.0                                                          QS046580
      SUM7=0.0                                                          QS046590
      DO 56 J=1,MUM                                                     QS046600
      A=ZCAM(J)                                                         QS046610
      B=ZCAM(J+1)                                                       QS046620
      A1=ABS(REAL(B-A))                                                 QS046630
      A2=ABS(AIMAG(B-A))                                                QS046640
      IF(A1 .GE. A2) SUM2=SUM2+BINF(I,J)*REAL(STREN(J))                 QS046650
      IF(A1 .LT. A2) SUM2=SUM2+BINF(I,J)*AIMAG(STREN(J))                QS046660
      FUN1=AIMAG(SOUR(I,J))-REAL(SOUR(I,J))*AIMAG(B-A)/REAL(B-A)        QS046670
      SUM7=SUM7+REAL(STREN(J))*FUN1                                     QS046680
56    CONTINUE                                                          QS046690
      ZETB(I)=SUM7                                                      QS046700
55    CONTINUE                                                          QS046710
      CHORX=CHORD                                                       QS046720
      DELX=3.0*CHORX/(NUMX-1.0)                                         QS046730
      DO 60 I=1,1                                                       QS046740
      DO 60 J=1,NUMY                                                    QS046750
      X(I,J)=-2.0*PSAVE                                                 QS046760
60    CONTINUE                                                          QS046770
      DELY=PSAVE/(NUMY-2)                                               QS046780
      DO 70 I=1,1                                                       QS046790
      DO 70 J=1,NUMY                                                    QS046800
      Y(I,J)=DELY*(J-2)                                                 QS046810
      IF(J .EQ. 1) Y(I,J)=-0.00001                                      QS046820
      IF(J .EQ. 2) Y(I,J)=0.00001                                       QS046830
      SUM=0.0                                                           QS046840
      Z=X(I,J)+AI*Y(I,J)                                                QS046850
      IF(X(I,J) .LT. 0.0 .OR. X(I,J) .GT. XMAX) GO TO 7031              QS046860
      DO 7060 L=1,IMAX                                                  QS046870
      IF(X(I,J) .GE. REAL(ZCAM(L)) .AND. X(I,J) .LE.  REAL(ZCAM(L+1))) GQS046880
     *O TO 7080                                                         QS046890
7060  CONTINUE                                                          QS046900
7080  CONTINUE                                                          QS046910
      L1=L                                                              QS046920
      DO 7100 L=IMAX,NUM                                                QS046930
      IF(X(I,J) .LE. REAL(ZCAM(L)) .AND. X(I,J) .GE. REAL(ZCAM(L+1))) GOQS046940
     * TO 7120                                                          QS046950
7100  CONTINUE                                                          QS046960
7120  CONTINUE                                                          QS046970
      L2=L                                                              QS046980
      YA=AIMAG(ZCAM(L1))                                                QS046990
      YB=AIMAG(ZCAM(L1+1))                                              QS047000
      XA=REAL(ZCAM(L1))                                                 QS047010
      XB=REAL(ZCAM(L1+1))                                               QS047020
      Y1=YA+(YB-YA)/(XB-XA)*(X(I,J)-XA)                                 QS047030
      YA=AIMAG(ZCAM(L2))                                                QS047040
```

```
      YB=AIMAG(ZCAM(L2+1))
      XA=REAL(ZCAM(L2))
      XB=REAL(ZCAM(L2+1))
      Y2=YA+(YB-YA)/(XB-XA)*(X(I,J)-XA)
 7031 CONTINUE
      DO 75 IX=1,MUM
      A=ZCAM(IX)
      B=ZCAM(IX+1)
      ZA=CLOG(PI/PITCH*(Z-A))
      ZB=CLOG(PI/PITCH*(Z-B))
      IF(X(I,J) .GT. 0.0) GO TO 7000
      IFLA=1
      IF(Y(I,J) .GE. 0.0) GO TO 7010
      ZA=ZA
      ZB=ZB
      GO TO 7020
 7010 CONTINUE
      IF(AIMAG(ZA) .LT. 0.0) ZA=ZA+2.0*AI*PI
      IF(AIMAG(ZB) .LT. 0.0) ZB=ZB+2.0*AI*PI
      GO TO 7020
 7000 CONTINUE
      IF(X(I,J) .GT. XMAX) GO TO 7040
      IF(Y(I,J) .LT. Y1 .AND. Y(I,J) .GT. Y2) IFLA=2
      IF(Y(I,J) .LT. Y1) GO TO 7140
      IF(AIMAG(ZA) .LT. -PI/2.0) ZA=ZA+2.0*AI*PI
      IF(AIMAG(ZB) .LT. -PI/2.0) ZB=ZB+2.0*AI*PI
      GO TO 7020
 7140 CONTINUE
      IF(Y(I,J) .LE. Y2) GO TO 7160
      IF(AIMAG(ZA) .LT. 0.0 .AND. IX .GE. IMAX) ZA=ZA+2.0*AI*PI
      IF(AIMAG(ZB) .LT. 0.0 .AND. IX .GE. IMAX) ZB=ZB+2.0*AI*PI
      GO TO 7020
 7160 CONTINUE
      ZA=ZA
      ZB=ZB
      IF(IX .GE. IMAXY .AND. AIMAG(ZA) .GT. PI/2.0) ZA=ZA-2.0*PI*AI
      IF(IX .GE. IMAXY .AND. AIMAG(ZB) .GT. PI/2.0) ZB=ZB-2.0*PI*AI
      GO TO 7020
 7040 CONTINUE
      IFLA=1
      ZA=ZA
      ZB=ZB
 7020 CONTINUE
 5000 CONTINUE
      SUM=SUM+STREN(IX)*(A-B+(Z-A)*ZA-(Z-B)*ZB)
      ZO=(A+B)/2.0
      ZTEL=(Z-ZO)*PI/PITCH
      HTAN=(CEXP(ZTEL)-CEXP(-ZTEL))/(CEXP(ZTEL)+CEXP(-ZTEL))
```

```
      SUM=SUM+STREN(IX)*CLOG(HTAN/ZTEL)*(B-A)
      IF(I .GT. 1 .AND. I .LT. NUMX) GO TO 3010
      SUM3=(A-B)+(Z-A)*ZA-(Z-B)*ZB
      SUM3= SUM3*STREN(IX)
      SUM4=CLOG(HTAN/ZTEL)*(B-A)
      SUM4= SUM4*STREN(IX)
      SUM5=CLOG(1.0/ZTEL)*(B-A)
      SUM5= SUM5*STREN(IX)
      SUM6=CLOG(HTAN)*(B-A)
      SUM6= SUM6*STREN(IX)
3010  CONTINUE
75    CONTINUE
      ETA(I,J)=REAL(SUM)
      IF(X(I,J) .LT.  XMAX .AND. X(I,J) .GT. 0.0 .AND. Y(I,J) .LT. Y1 .
     *AND
     1.Y(I,J) .GT. Y2) ETA(I,J)=1.05
      ZETA(I,J)=AIMAG(SUM)
70    CONTINUE
      NUMD=NUMY-1
      DO 310 ID=1,NUMD
      ARG(ID)=Y(1,ID+1)
      FUN(ID)=ETA(1,ID+1)
      GUN1(ID)=ZETA(1,ID+1)
310   CONTINUE
      FO=0.0
      YSTAR2=.95*PSAVE/2.
      TEST=XNTRP(ARG,FUN,FO,NUMD,YSTAR2)
      ZETA2=FNTRP(ARG,GUN1,YSTAR2,NUMD)
      ETEST=0.0
      ITEST=1
      DO 330 I=1,NUMD
      IF(ETEST .LE. ETA(1,I+1)) ETEST=ETA(1,I+1)
      IF(ETEST .LE. ETA(1,I+1)) ITEST=I
330   CONTINUE
      IF(ITEST .EQ. 1) ITEST=2
      AQ1=ZETA(1,ITEST+1)
      DELT=Y(1,4)-Y(1,2)
      AQ2=(ZETA(1,ITEST+2)-ZETA(1,ITEST))/(2.0*DELT)
      AQ3=(ZETA(1,ITEST+2)+ZETA(1,ITEST)-2.0*ZETA(1,ITEST+1))/(2.0*DELT*
     **2)
      BQ2=(ETA(1,ITEST+2)-ETA(1,ITEST))/(2.0*DELT)
      BQ3=(ETA(1,ITEST+2)+ETA(1,ITEST)-2.0*ETA(1,ITEST+1))/(2.0*DELT**2)
      YLMAX=-BQ2/(2.0*BQ3)
      ZETA4=AQ1+AQ2*YLMAX+AQ3*YLMAX**2
      DELZ=ZETA(1,2)-ZETA(1,1)
      DO 360 I=1,MUM
      ARG(I)=ZETB(I)
      FUN(I)=REAL(ZBODY(I))
```

```
360   CONTINUE                                          QS048010
      XBOD1=FNTRP(ARG,FUN,ZETA4,MUM)                    QS048020
      XBOD3=FNTRP(ARG,FUN,0.0,MUM)                      QS048030
      DO 370 I=1,MUM                                    QS048040
      FUN(I)=AIMAG(ZBODY(I))                            QS048050
370   CONTINUE                                          QS048060
      YBOD1=FNTRP(ARG,FUN,ZETA4,MUM)                    QS048070
      YBOD3=FNTRP(ARG,FUN,0.0,MUM)                      QS048080
      ILIM=MUM+2                                        QS048090
      XQ(1,1)=XBOD1                                     QS048100
      XQ(1,ILIM)=XBOD1                                  QS048110
      YQ(1,1)=YBOD1                                     QS048120
      YQ(1,ILIM)=YBOD1                                  QS048130
      ZQ(1,1)=ZETA4                                     QS048140
      ZQ(1,ILIM)=-ZETA4                                 QS048150
      EQ(1,1)=1.0                                       QS048160
      EQ(1,ILIM)=1.0                                    QS048170
      DO 380 I=1,MUM                                    QS048180
      IF(ZETB(I) .GT. ZETA4)  GO TO 390                 QS048190
380   CONTINUE                                          QS048200
390   CONTINUE                                          QS048210
      ISTART=I                                          QS048220
      DO 410 I=ISTART,MUM                               QS048230
      IL=I-ISTART+2                                     QS048240
      XQ(1,IL)=REAL(ZBODY(I))                           QS048250
      YQ(1,IL)=AIMAG(ZBODY(I))                          QS048260
      ZQ(1,IL)=ZETB(I)                                  QS048270
      EQ(1,IL)=1.0                                      QS048280
410   CONTINUE                                          QS048290
      IEND=ISTART-1                                     QS048300
      DO 420 I=1,IEND                                   QS048310
      IL=MUM-ISTART+2+I                                 QS048320
      XQ(1,IL)=REAL(ZBODY(I))                           QS048330
      YQ(1,IL)=AIMAG(ZBODY(I))                          QS048340
      ZQ(1,IL)=ZETB(I)-DELZ                             QS048350
      EQ(1,IL)=1.0                                      QS048360
420   CONTINUE                                          QS048370
                                                        QS048380
                                                        QS048390
C      DETERMINE LOCATION OF PERIODIC BOUNDARY POINTS   QS048400
                                                        QS048410
      DO 110 IL=2,4                                     QS048420
      IF(IL .EQ. 2) YQ(2,1)=YSTAR2                      QS048430
      IF(IL .EQ. 2) XQ(2,1)=-1.50*PSAVE                 QS048440
      IF(IL .EQ. 2) EQ(2,1)=0.0                         QS048450
      IF(IL .EQ. 2) ZQ(2,1)=ZETA2                       QS048460
      IF(IL .EQ. 2) DELX=(CHORX+3.*PSAVE)/(NETA0-1.0)   QS048470
      IF(IL .EQ. 2) IEND=NETA0                          QS048480
```

```
      IF(IL .EQ. 2) ETAO=0.0                                            QS048490
      IF(IL .EQ. 3) YQ(3,1)=YBOD3                                       QS048500
      IF(IL .EQ. 3) XQ(3,1)=XBOD3                                       QS048510
      IF(IL .EQ. 3) EQ(3,1)=1.0                                         QS048520
      IF(IL .EQ. 3) ZQ(3,1)=0.0                                         QS048530
      IF(IL .EQ. 3) DELX=(2.0*PSAVE)/(NZETO-1.0)                        QS048540
      IF(IL .EQ. 3) IEND=NZETO                                          QS048550
      IF(IL .EQ. 3) ZETAO=0.0                                           QS048560
      IF(IL .EQ. 4) YQ(4,1)=YBOD1                                       QS048570
      IF(IL .EQ. 4) XQ(4,1)=XBOD1                                       QS048580
      IF(IL .EQ. 4) EQ(4,1)=1.0                                         QS048590
      IF(IL .EQ. 4) ZQ(4,1)=ZETA4                                       QS048600
      IF(IL .EQ. 4) DELX=(-2.0*PSAVE-XBOD1)/(NZET4-1.0)                 QS048610
      IF(IL .EQ. 4) IEND=NZET4                                          QS048620
      IF(IL .EQ. 4) ZETAO=ZETA4                                         QS048630
      DO 110 I=2,IEND                                                   QS048640
      XQ(IL,I)=XQ(IL,I-1)+DELX                                          QS048650
      YQ(IL,I)=YQ(IL,I-1)                                               QS048660
      DO 120 ITOL=1,200                                                 QS048670
      SUM=0.0                                                           QS048680
      SUMD=0.0                                                          QS048690
      Z=XQ(IL,I)+AI*YQ(IL,I)                                            QS048700
      IF(XQ(IL,I) .LT. 0.0 .OR. XQ(IL,I) .GT. XMAX) GO TO 130           QS048710
      DO 140 L=1,IMAX                                                   QS048720
      IF(XQ(IL,I) .GE. REAL(ZCAM(L)) .AND. XQ(IL,I) .LE. REAL(ZCAM(L+1))QS048730
     *) GO TO 150                                                       QS048740
  140 CONTINUE                                                          QS048750
  150 CONTINUE                                                          QS048760
      L1=L                                                              QS048770
      DO 160 L=IMAX,NUM                                                 QS048780
      IF(XQ(IL,I) .LE. REAL(ZCAM(L)) .AND. XQ(IL,I) .GE. REAL(ZCAM(L+1))QS048790
     *) GO TO 170                                                       QS048800
  160 CONTINUE                                                          QS048810
  170 CONTINUE                                                          QS048820
      L2=L                                                              QS048830
      YA=AIMAG(ZCAM(L1))                                                QS048840
      YB=AIMAG(ZCAM(L1+1))                                              QS048850
      XA=REAL(ZCAM(L1))                                                 QS048860
      XB=REAL(ZCAM(L1+1))                                               QS048870
      Y1=YA+(YB-YA)/(XB-XA)*(XQ(IL,I)-XA)                               QS048880
      YA=AIMAG(ZCAM(L2))                                                QS048890
      YB=(AIMAG(ZCAM(L2+1)))                                            QS048900
      XA=REAL(ZCAM(L2))                                                 QS048910
      XB=REAL(ZCAM(L2+1))                                               QS048920
      Y2=YA+(YB-YA)/(XB-XA)*(XQ(IL,I)-XA)                               QS048930
  130 CONTINUE                                                          QS048940
      DO 180 IX=1,MUM                                                   QS048950
      A=ZCAM(IX)                                                        QS048960
```

```
      B=ZCAM(IX+1)                                                      QS048970
      ZA=CLOG(PI/PITCH*(Z-A))                                           QS048980
      ZB=CLOG(PI/PITCH*(Z-B))                                           QS048990
      IF(XQ(IL,I) .GT. 0.0) GO TO 190                                   QS049000
      IF(YQ(IL,I) .GE.  0.0) GO TO 200                                  QS049010
      ZA=ZA                                                             QS049020
      ZB=ZB                                                             QS049030
      GO TO 210                                                         QS049040
  200 CONTINUE                                                          QS049050
      IF(AIMAG(ZA) .LT. 0.0) ZA=ZA+2.0*AI*PI                            QS049060
      IF(AIMAG(ZB) .LT. 0.0) ZB=ZB+2.0*AI*PI                            QS049070
      GO TO 210                                                         QS049080
  190 CONTINUE                                                          QS049090
      IF(XQ(IL,I) .GT. XMAX) GO TO 220                                  QS049100
      IF(YQ(IL,I) .LT. Y1) GO TO 230                                    QS049110
      IF(AIMAG(ZA) .LT. -PI/2.0) ZA=ZA+2.0*AI*PI                        QS049120
      IF(AIMAG(ZB) .LT. -PI/2.0) ZB=ZB+2.0*AI*PI                        QS049130
      GO TO 210                                                         QS049140
  230 CONTINUE                                                          QS049150
      IF(YQ(IL,I) .LE. Y2) GO TO 240                                    QS049160
      IF(AIMAG(ZA) .LT. 0.0 .AND. IX .GE. IMAX) ZA=ZA+2.0*AI*PI         QS049170
      IF(AIMAG(ZB) .LT. 0.0 .AND. IX .GE. IMAX) ZB=ZB+2.0*AI*PI         QS049180
      GO TO 210                                                         QS049190
  240 CONTINUE                                                          QS049200
      ZA=ZA                                                             QS049210
      ZB=ZB                                                             QS049220
      IF(IX .GE. IMAXY .AND . AIMAG(ZA) .GT. PI/2.0) ZA=ZA-2.0*PI*AI    QS049230
      IF(IX .GE. IMAXY .AND. AIMAG(ZB) .GT. PI/2.0) ZB=ZB-2.0*PI*AI     QS049240
      GO TO 210                                                         QS049250
  220 CONTINUE                                                          QS049260
      ZA=ZA                                                             QS049270
      ZB=ZB                                                             QS049280
  210 CONTINUE                                                          QS049290
      SUM=SUM+STREN(IX)*(A-B+(Z-A)*ZA-(Z-B)*ZB)                         QS049300
      SUMD=SUMD+STREN(IX)*(ZA-ZB)                                       QS049310
      ZO=(A+B)/2.0                                                      QS049320
      ZTEL=(Z-ZO)*PI/PITCH                                              QS049330
      HTAN=(CEXP(ZTEL)-CEXP(-ZTEL))/(CEXP(ZTEL)+CEXP(-ZTEL))            QS049340
      HSIN=0.5*(CEXP(ZTEL*2.0)-CEXP(-ZTEL*2.0))                         QS049350
      SUM=SUM+STREN(IX)*CLOG(HTAN/ZTEL)*(B-A)                           QS049360
      SUMD=SUMD+STREN(IX)*(PI/PSAVE/HSIN-1.0/(Z-ZO))*(B-A)              QS049370
  180 CONTINUE                                                          QS049380
      ETAF=REAL(SUM)                                                    QS049390
      ZETAF=AIMAG(SUM)                                                  QS049400
      ETAFD=AIMAG(SUMD)                                                 QS049410
      ZETAFD=-REAL(SUMD)                                                QS049420
      F1=ETAO-ETAF                                                      QS049430
      G1=ZETAO-ZETAF                                                    QS049440
```

```
      DELY1=-F1/ETAFD                                                   QS049450
      DELY2=-G1/ZETAFD                                                  QS049460
      IF(IL .EQ. 2) DELT=DELY1                                          QS049470
      IF(IL .GT. 2) DELT=DELY2                                          QS049480
      YQ(IL,I)=YQ(IL,I)+DELT                                            QS049490
      IF(ABS(DELT) .LT. 1.E-06) GO TO 250                               QS049500
  120 CONTINUE                                                          QS049510
  250 CONTINUE                                                          QS049520
      IF(IL .EQ. 2) EQ(IL,I)=0.0                                        QS049530
      IF(IL .EQ. 2) ZQ(IL,I)=ZETAF                                      QS049540
      IF(IL .EQ. 3) EQ(IL,I)=ETAF                                       QS049550
      IF(IL .EQ.3) ZQ(IL,I)=0.0                                         QS049560
      IF(IL .EQ. 4) EQ(IL,I)=ETAF                                       QS049570
      IF(IL .EQ. 4) ZQ(IL,I)=ZETA4                                      QS049580
  110 CONTINUE                                                          QS049590
                                                                        QS049600
                                                                        QS049610
                                                                        QS049620
                                                                        QS049630
      NEND= 2*NZGRID-1                                                  QS049640
      DO 500 I=1,NEND                                                   QS049650
      DO 500 J=1,NEGRID                                                 QS049660
      ETA(I,J)=0.0                                                      QS049670
      ZETA(I,J)=0.0                                                     QS049680
      X(I,J)=0.0                                                        QS049690
      Y(I,J)=0.0                                                        QS049700
  500 CONTINUE                                                          QS049710
      DELZ=ZQ(4,1)/(NZGRID-1.0)                                         QS049720
      DELE=1.0/(NEGRID-1.0)                                             QS049730
      DO 510 I=1,NETA0                                                  QS049740
      ARG(I)=ZQ(2,I)                                                    QS049750
      FUN(I)=XQ(2,I)                                                    QS049760
      GUN1(I)=YQ(2,I)                                                   QS049770
  510 CONTINUE                                                          QS049780
      DO 520 J=1,NZGRID                                                 QS049790
      ETA(J,1)=0.0                                                      QS049800
      ZETA(J,1)=ZG(4,1)-DELZ*(J-1.0)                                    QS049810
      RQ=ZETA(J,1)                                                      QS049820
      X(J,1)=FNTRP(ARG,FUN,RQ,NETA0)                                    QS049830
      Y(J,1)=FNTRP(ARG,GUN1,RQ,NETA0)                                   QS049840
  520 CONTINUE                                                          QS049850
      NEND=2*NZGRID-1                                                   QS049860
      ISTART=NZGRID+1                                                   QS049870
      DO 530 J=ISTART,NEND                                              QS049880
      IML=2*NZGRIC-J                                                    QS049890
      ETA(J,1)=0.0                                                      QS049900
      ZETA(J,1)=-ZETA(IML,1)                                            QS049910
      X(J,1)=X(IML,1)                                                   QS049920
```

```
      Y(J,1)=Y(IML,1)-PSAVE                                             QS049930
  530 CONTINUE                                                          QS049940
      ILIM=MUM+2                                                        QS049950
      DO 540 I=1,ILIM                                                   QS049960
      ARG(I)=ZQ(1,I)                                                    QS049970
      FUN(I)=XQ(1,I)                                                    QS049980
      GUN1(I)=YQ(1,I)                                                   QS049990
  540 CONTINUE                                                          QS050000
      DO 550 J=1,NEND                                                   QS050010
      ETA(J,NEGRID)=1.0                                                 QS050020
      ZETA(J,NEGRID)=ZETA(J,1)                                          QS050030
      RQ=ZETA(J,NEGRID)                                                 QS050040
      X(J,NEGRID)=FNTRP(ARG,FUN,RQ,ILIM)                                QS050050
      Y(J,NEGRID)=FNTRP(ARG,GUN1,RQ,ILIM)                               QS050060
  550 CONTINUE                                                          QS050070
      DO 560 I=1,NZET4                                                  QS050080
      IML=NZET4+1-I                                                     QS050090
      ARG(I)=EQ(4,IML)                                                  QS050100
      FUN(I)=XQ(4,IML)                                                  QS050110
      GUN1(I)=YQ(4,IML)                                                 QS050120
  560 CONTINUE                                                          QS050130
      DO 570 I=1,NEGRID                                                 QS050140
      ETA(1,I)=(I-1.0)*DELE                                             QS050150
      ZETA(1,I)=ZQ(4,1)                                                 QS050160
      RQ=ETA(1,I)                                                       QS050170
      X(1,I)=FNTRP(ARG,FUN,RQ,NZET4)                                    QS050180
      Y(1,I)=FNTRP(ARG,GUN1,RQ,NZET4)                                   QS050190
  570 CONTINUE                                                          QS050200
      DO 580 I=1,NEGRID                                                 QS050210
      ETA(NEND,I)=ETA(1,I)                                              QS050220
      ZETA(NEND,I)=-ZETA(1,I)                                           QS050230
      X(NEND,I)=X(1,I)                                                  QS050240
      Y(NEND,I)=Y(1,I)                                                  QS050250
  580 CONTINUE                                                          QS050260
      DO 585 I=1,NZETO                                                  QS050270
      IML=NZETO+1-I                                                     QS050280
      ARG(I)=EQ(3,IML)                                                  QS050290
      FUN(I)=XQ(3,IML)                                                  QS050300
      GUN1(I)=YQ(3,IML)                                                 QS050310
  585 CONTINUE                                                          QS050320
      DO 587 I=1,NEGRID                                                 QS050330
      ETA(NZGRID,I)=(I-1.0)*DELE                                        QS050340
      ZETA(NZGRID,I)=0.0                                                QS050350
      RQ=ETA(NZGRID,I)                                                  QS050360
      X(NZGRID,I)=FNTRP(ARG,FUN,RQ,NZETO)                               QS050370
      Y(NZGRID,I)=FNTRP(ARG,GUN1,RQ,NZETO)                              QS050380
  587 CONTINUE                                                          QS050390
      X(1,1)=-4.0*PSAVE                                                 QS050400
```

```
      X(NZGRID,1)=4.0*PSAVE
      X(NEND,1)=-4.0*PSAVE
      Y(1,1)=YQ(4,NZET4)
      Y(NZGRID,1)=YQ(3,NZETO)
      Y(NEND,1)=YQ(4,NZET4)

      L=NEGRID
      L=NEND
      L=NZGRID

C      GIVEN PERIODIC BOUNDARY AND BLADE SHAPE, SOLVE THE BOUNDARY VALUE
C      PROBLEM
C      FOR THE INTERNAL GRID POINTS.

      BIJ=2.0*(1.0/(DELE**2)+1.0/(DELZ**2))
      BIJP=1.0/(DELE**2)
      BIJM=BIJP
      BIPJ=1.0/(DELZ**2)
      BIMJ=BIPJ
      NEG2=NEGRID-1
      NZG2=NEND-1
      EPP=1.E-05
      DO 600 ITER=1,100
      ERR=0.0
      DO 610 I=2,NZG2
      DO 610 J=2,NEG2
      IF(I .EQ. NZGRID) GO TO 610
      XHOLD=X(I,J)
      YHOLD=Y(I,J)
      X(I,J)=C/BIJ*(BIPJ*X(I+1,J)+BIMJ*X(I-1,J)+BIJP*X(I,J+1)+BIJM*X(I
     *,J-1)) +(1.-C)*X(I,J)
      Y(I,J)=C/BIJ*(BIPJ*Y(I+1,J)+BIMJ*Y(I-1,J)+BIJP*Y(I,J+1)+BIJM*Y(I
     *,J-1)) +(1.-C)*Y(I,J)
      ERR=ERR+ABS(X(I,J)-XHOLD)+ABS(Y(I,J)-YHOLD)
610   CONTINUE
      IF(ITER .LT. 5) GO TO 600
      ERR=ERR/(2.0*NEG2*NZG2)
      IF(ERR .LE. EPP) GO TO 670
      KPRT=ITER/200
      IF(ITER .NE. 200*KPRT) GO TO 600
      DO 620 I=1,NEND
620   CONTINUE
600   CONTINUE



670   CONTINUE
```

```
      WRITE(6,4444) ITER,ERR
4444  FORMAT(' ITER COUNT FOR INTERNAL GRID POINTS= ',I3,' ERR= ',E15.5)
      DO 650 I=1,NEND
      DO 650 J=1,NEGRID
      ZPET(I,J)=X(I,J)+AI*Y(I,J)
650   CONTINUE


      MM=NEGRID
      NN=NEND
      K=1
      ZPET(1,1)=-2.0*CHORB+AI*YQ(4,NZET4)
      ZPET(NZGRID,1)=2.0*CHORB+AI*YQ(3,NZETO)
      ZPET(NEND,1)=-2.0*CHORB+AI*YQ(4,NZET4)
C     ZPET(1,1)=-2.*CHORX+0.
C     ZPET(NEND,1)=ZPET(1,1)
C     ZPET(NZGRID,1)=2.*CHORX+0.


      DO 675 I=1,NEND
      DO 675 J=1,NEGRID
      ZPET(I,J)=ZPET(I,J)*CEXP(-AI*STAG*PI/180.0)
      X(I,J)=REAL(ZPET(I,J))
      Y(I,J)=AIMAG(ZPET(I,J))
675   CONTINUE
      ETADF1=-DELZ
      ETADF2=-ETADF1
      ZETDF1= DELE
      ZETDF2=-ZETDF1
      ET(1)=FLOAT(NZGRID-1)*(-DELZ)
      ZET(1)=0.
      DO 4844 I1=2,NEND
4844  ET(I1)=DELZ+ET(I1-1)
      DO 4845 I1=2,NEGRID
4845  ZET(I1)=DELE+ZET(I1-1)
      NOZ=NEND
      DO 780 I=1,NEND
      DO 780 J=1,NEGRID
      ZPET(I,J)=ZPET(I,J)*CEXP(AI*STAG*PI/180.0)
780   CONTINUE
      DO 790 I=1,NEND
      ZTEMP=(ZPET(I,NEGRID)+ZPET(I+1,NEGRID))/2.0
      ZTEMQ=ZTEMP*CEXP(-AI*STAG*PI/180.0)
      XT=REAL(ZTEMP)
      YT=AIMAG(ZTEMP)
      XR=REAL(ZTEMQ)
      YR=AIMAG(ZTEMQ)
790   CONTINUE
```

```
      STAG=STAG*PI/180.
      PITCH=PSAVE
      RETURN
      END
      SUBROUTINE CHLSKY(A,C,N)
      IMPLICIT REAL*8 (A-H,O-Z)
C - CHOLESKY - SOLVES AN N BY N SYSTEM OF EQUATIONS WITH COEFFICIENT
C     -
C    MATRIX A(I,J), CONSTANT VECTOR C(I), AND SOLUTION VECTOR.
C    CHOLESKYS METHOD IS USED.  IT REQUIRES N**2 + O(N) OPERATIONS.
C
      DIMENSION A(63,63),C(63)
C
C - CALCULATION OF SOLUTION MATRIX.  NOTE BOTH THE UPPER AND LOWER
C    TRIANGULAR MATRICES ARE STORED AS REPLACEMENT VALUES IN MATRIX A.
C    ALSO THE SOLUTION VECTOR REPLACES THE CONSTANT VECTOR (C).
C

      IST = 1
      JST = 2
      DO 50 K = 1,N
C
C - CALCULATION OF LOWER TRIANGULAR MATRIX ELEMENTS
C
      DO 10 I = IST,N
      J = 0
20    J = J+1
      IF (J.GE.K) GO TO 10
      A(I,K) = A(I,K) - A(I,J)*A(J,K)
      GO TO 20
10    CONTINUE
C
C - CALCULATION OF UPPER TRIANGULAR MATRIX ELEMENTS
C
      IF (K.EQ.N) GO TO 50
      DO 40 J = JST,N
      I = 0
30    I = I+1
      IF (I.GE.K) GO TO 40
      A(K,J) = A(K,J)-A(K,I)*A(I,J)
      GO TO 30
40    A(K,J) = A(K,J)/A(K,K)
      IST = IST+1
50    JST = JST+1
C
C - CALCULATION OF CONSTANT VECTOR ELEMENTS.  ENTRY POINT FOR MULTIPLE
C    SOLUTION INVOLVING THE SAME COEFFICIENT MATRIX, BUT CHANGES IN
C    CONSTANT MATRIX.
```

```
C                                                                       QS051850
      ENTRY CHGCNT(A,C,N)                                               QS051860
      NP1 = N+1                                                         QS051870
      DO 70 K = 1,N                                                     QS051880
      I = 0                                                             QS051890
60    I = I+1                                                           QS051900
      IF (I.GE.K) GO TO 70                                              QS051910
      C(K) = C(K) - A(K,I)*C(I)                                         QS051920
      GO TO 60                                                          QS051930
70    C(K) = C(K)/A(K,K)                                                QS051940
C                                                                       QS051950
C - EVALUATION OF SOLUTION MATRIX                                       QS051960
C                                                                       QS051970
      DO 90 K = 1,N                                                     QS051980
      L = NP1-K                                                         QS051990
      M = NP1                                                           QS052000
80    M = M-1                                                           QS052010
      IF (M.LE.L) GO TO 90                                              QS052020
      C(L) = C(L)-A(L,M)*C(M)                                           QS052030
      GO TO 80                                                          QS052040
90    CONTINUE                                                          QS052050
      RETURN                                                            QS052060
      END                                                               QS052070
```

LIST OF REFERENCES

1. NASA Report SP-36, Aerodynamic Design of Axial-Flow Compressors, edited by Irving A. Johnsen and Robert A. Bullock, 1965.

2. NACA Report 1016, Effect of Tunnel Configuration and Testing Technique on Cascade Performance, by John R. Erwin and James C. Emery, 1951.

3. Duval, David A., Evaluation of a Subsonic Cascade Wind Tunnel for Compressor Blade Testing, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1980.

4. Rose, C. and Guttormson, D. L., Installation and Test of a Rectilinear Cascade, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1964.

5. Bartocci, J. E., An Investigation of the Flow Conditions at the Lower Measuring Plane, and in the Plenum Chamber of the Rectilinear Cascade Test Facility, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1966.

6. Moebius, Richard C., Analysis and Testing to Improve the Flow from the Plenum of a Subsonic Cascade Wind Tunnel, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1980.

7. Cina, Frank S., Subsonic Cascade Wind Tunnel Tests Using a Compressor Configuration of DCA Blades, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1981.

8. NASA Technical Note TN D-5427, Fortran Program for Calculating Transonic Velocities on a Blade-to-Blade Stream Surface of a Turbomachine, by Theodore Katsanis, 1969.

9. NASA Handout, Draft of Technical Paper, Quasi-3D Full Potential Transonic Blade-to-Blade Code, by Charles Farrell, 1981, Conference on Turbomachinery Flow Analysis Methods--A Status Report on Maturing Codes, Conference at NASA Lewis Research Center, 14-15 October 1981.

10. NASA Technical Paper 1493, Performance of Two-Stage Fan Having Low-Aspect-Ratio First-Stage Rotor Blading, by Donald C. Urasek, William T. Gorrell and Walter S. Cunnan, 1979.

11. 3052A System Library (9845A), Hewlett-Packard Company, 1978.

12. Turbopropulsion Laboratory, Naval Postgraduate School, Technical Note 80-03, Data Acquisition Programs for the Subsonic Cascade Wind Tunnel, by D. A. Duval, August 1980.

13. W. R. Church Computer Center, Naval Postgraduate School, Technical Note VM-01, User's Guide to VM/CMS at NPS, October 1981.

14. W. R. Church Computer Center, Naval Postgraduate School, Technical Note VM-05, Introduction to the XEDIT EDITOR, May 1981.

15. NASA Technical Note TN D-8431, Revised Fortran Program for Calculating Velocities and Streamlines on the Hub-Shroud Midchannel Stream Surface of an Axial, Radial, or Mixed-Flow Turbomachine or Annular Duct, by Theodore Katsanis and William D. McNally, 1977.

16. Turbopropulsion Laboratory, Naval Postgraduate School, Project Report NPS67-81-019PR, Uniform Inlet Conditions for the NPS Subsonic Cascade Wind Tunnel, by Donald M. McEligot, December 1981.

17. Pankhurst, R. C. and Holder, D. W., Wind Tunnel Technique, Pitman and Sons, 1952.

18. NASA Technical Paper TP-2030, Computer Program for Calculating Full Potential Transonic Quasi-Three-Dimensional Flow Through a Rotating Turbomachinery Blade Row, by Charles Farrell, 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Director, Turbopropulsion Laboratory, Code 67Sf Naval Postgraduate School Monterey, California 93940	15
5.	Fan and Compressor Branch (ATTN: N. Sanger) Mail Stop 5-9 NASA Lewis Research Center 21000 Brookpark Road Cleveland, Ohio 44135	2
6.	LCDR W. D. Molloy Jr. VC-5, NAS Cubi Point FPO San Francisco, California 96601	2